The End of End-to-End Security?

In 1984, a year after the Arpanet switched from using the Network Control Protocol to using the TCP/IP protocol suite, Jerry Saltzer, David Reed, and Dave Clark expressed what became the core concept of the Internet in a short paper titled "End-to-End Arguments in System Design."[1]

SCOTT BRADNER
Harvard University

More than two decades following the philosophy in this paper explains both the Internet's dynamic generative effect and the mess that many observers see when the discussion turns to Internet security.

The end-to-end principle

The end-to-end arguments paper presents a "design principle" that one should not place mechanisms in the network if they can be placed in end nodes; thus, networks should provide general services rather than services that are designed to support specific applications. The design and implementation of the Internet followed this design principle well. The Internet was designed to be an application-agnostic datagram delivery service. The Internet of today isn't as pure an implementation of the end-to-end design principle as it once was, but it's enough of one that the collateral effects of the network not knowing what's running over it are becoming major problems, at least in the minds of some observers.

Before I get to those perceived problems, I'd like to talk about what the end-to-end design principle has meant to the Internet, technical evolution, and society.

The Internet doesn't care what you do—its job is just to "deliver the bits, stupid" (in the words of David Isenberg in his 1997 paper, "Rise of the Stupid Network"[2]). The "bits" could be part of an email message, a data file, a photograph, or a video, or they could be part of a denial-of-service attack, a malicious worm, a break-in attempt, or an illegally shared song. The Net doesn't care, and that is both its power and its threat.

The Internet (and by this, I mean the Arpanet, the NSFNet, and the networks of their successor commercial ISPs) wasn't designed to run the World Wide Web. The Internet wasn't designed to run Google Earth. It was designed to support them even though they did not exist at the time the foundations of the Net were designed. It was designed to support them by being designed to transport data without caring what it was that data represented.

At the very first, the design of TCP/IP wasn't so flexible. The initial design had TCP and IP within a single protocol, one that would only deliver data reliably to a destination. But it was realized that not all applications were best served by a protocol that could only deliver reliable data streams.[3] In particular, timely delivery of information is more important than reliable delivery when trying to support interactive voice over a network if adding reliability would, as it does, increase delay. TCP was split from IP so that the application running in an end node could determine for itself the level of reliability it needed. This split created the flexibility that is currently being used to deliver Skype's interactive voice service over the same network that CNN uses to deliver up-to-the-minute news headlines and the US Patent and Trademark office uses to deliver copies of US patents.

Thus the Internet design, based as it was on the end-to-end principle, became a generative facility. Unlike the traditional phone system, in which most new applications must be installed in the phone switches deep in the phone network, anyone could create new applications and run them over the Internet without getting permission from the organizations that run the parts of the Net. This ability was exploited with "irrational exuberance"[4] during the late 1990s Internet boom. But, in spite of the hundreds of billions of dollars lost by investors when the boom busted, the number of Internet users and Web sites, the amount of Internet traffic, and the value of Internet commerce have continued to rise, and the rate of new ideas for Internet-based services hasn't noticeably diminished.

Opaqueness in the Net

Of course, my earlier description of a pure end-to-end Internet doesn't reflect current reality. Few Internet users live in such a world. At work, most users are behind corporate-run firewalls, which are specifically designed to get in the way of communications they aren't configured to understand. At home, many people who have broadband access have also installed network address translators (NATs) to deal with the fact that many ISPs permit only a single IP address per customer. NATs rewrite the addressing information in the packet headers and can block operation of some types of applications. For most Internet users, the end-to-end model is well on its way to being dead. Application designers, such as Vonage and Skype, are able to deal with the roadblocks in some cases, but their solutions often don't work through firewalls and can be complex if more than one user is behind a NAT.

To be a bit more precise, the end-to-end model at the network layer is at risk. But there is a layer where the end-to-end model still lives—HTTP. Most firewalls are configured to pass Web traffic (HTTP), and NATs know how to properly deal with TCP (over which HTTP runs). More and more new applications are showing up using HTTP as a transport. (Mark Gaynor and I present one [tongue-in-cheek] solution to the piecemeal implementation of this work-around.[5])

That said, most enterprises themselves do live in an end-to-end model world, given that ISPs generally don't filter traffic to and from enterprises, at least for now.

Security and privacy in an end-to-end world

Security isn't mentioned in Clark's paper, "The Design Philosophy of the DARPA Internet Protocols."[3] Nor is privacy. These were not goals of the design of the network itself. That's not to say that security and privacy weren't concerns. The end-to-end arguments paper used "secure transmission of data" as one reason that an end-to-end design was required.[1] The paper points out that network-level or per-link encryption doesn't actually provide assurance that a file that arrives at a destination is the same as the file that was sent or that the data went unobserved along the path from the source to the destination. The only way to ensure end-to-end data integrity and confidentiality is to use end-to-end encryption.

Thus, security and privacy are the responsibilities of the end nodes. If you want to ensure that a file will be transferred without any corruption, your data-transfer application had better include an integrity check, and if you didn't want to allow anyone along the way to see the data itself, your application had better encrypt it before transmitting it.

There are more aspects to security on a network than just data encryption. For example, to ensure that communication over the network is reliable, the network itself needs to be secure against attempts—purposeful or accidental—to disrupt its operation or redirect traffic away from its intended path. But the original Internet design didn't include protections against such attacks. Even if the network is working perfectly, you need to actually be talking to the server or person you think you are. But the Internet doesn't provide a way, at the network level, to assure the identities of its users or nodes. You also need to be sure that the message your computer receives isn't designed to exploit weaknesses in its software (such as worms or viruses) or in the ways that you use the Net. Protection against such things is the end system's responsibility.

Note that there is little that can be done "in the Net" or in your end system to protect your privacy from threats such as the government demanding the records of your use of Net-based services such as Google, which collect information about your network usage.

Many of today's observers assume that the lack of built-in protections against attacks and the lack of a secure way to identify users or nodes was a result of an environment of trust that prevailed when the original Internet design and protocols were developed. If you trusted the people on the Net, there was no need for special defensive functions. But a few people who were "at the scene" have told me that such protections were actively discouraged by the primary sponsor of the early Internet—that is to say, the US military wasn't all that interested in having good nonmilitary security, maybe because it might make its job harder in the future. Whatever the reason, the Internet wasn't designed to provide a secure environment that included protection against the malicious actions of those who would disrupt it or attack nodes or services provided over it.

The US military wasn't all that interested in having good nonmilitary security, maybe because it might make its job harder in the future.

The business of the Internet

Another feature of the end-to-end-based design is a lack of binding between the ISP that's providing your connectivity to the rest of the Internet and any Internet-based services you might want to use. You don't need to get your email service from your ISP; you can easily use Hotmail or Yahoo. You don't need to get your voice-over-IP (VoIP) service from your ISP; you can use Skype or Vonage. In both these cases, the ISP just has to deliver the bits. The characteristics of data delivery might have to be higher quality if you want to run VoIP rather than if you just want to send and receive email, but in neither case does your ISP know what you're doing unless you're getting one of those services from that ISP. But hundreds of thousands of Internet users have found that the characteristics of the normal data delivery they get from their ISP are just fine for VoIP most of the time.

Regulations that spring from these perceived needs are often in conflict with the end-to-end design principles that the Internet has been operating under.

Companies that have been in the ISP business for a while understand this—and they understand that when push comes to shove, they are in a commodity business: the ISP that can deliver the bits the cheapest with enough quality for the applications its users want to run will get the business. This does stress some ISPs' business models quite a bit, and, in many areas, the ISP business has emulated the airline business, in which prices are cut to well under what it costs to provide the service. It's possible to make money in a commodity business—as the "colored sugar water" (to use Steve Job's turn of phrase) vendors Coke and Pepsi have shown—but it can be hard to do for companies that don't compulsively focus on the cost of doing business.

The big new entrants in the ISP business are the telephone companies, and they have a very hard time understanding the concept that they might be running a commodity businesses and that they don't get revenue based on what application is running over the networks their customers buy from them. The leaders of both the new AT&T and Bell South have publicly said that they would like Google, Vonage and other providers of Internet services to pay them for the use of their wires. Given that, as I mentioned, many people find that such services work just fine over the user's current ISP service, the only way that ISPs might be able to convince Google to pay them is to make sure that the user gets bad Google service unless Google pays up.

Governments

Many national and local governments mostly ignored the Internet up until the past few years. Now many of them are trying to move the Internet under the same—or, in many cases, significantly more—regulations as the traditional telephone carriers have been under. Government justifications for the regulations range from protecting the consumer (for example, by making sure they get the service they expect or that they are protected from naughty pictures or viruses) and collecting taxes on Internet service and Internet commerce to universal service (making sure that everyone can get Internet service) and lawful intercept (that is, wiretapping). The regulators, by their basic nature, also tend to want to protect the incumbent telephone carriers. Other regulators want to be able to track communications between regulated groups, such as stockbrokers, and their clients.

Regulations that spring from these perceived needs often conflict with the end-to-end design principles that the Internet has operated under for most of its existence. The perceived need to ensure that users get a definable level of service leads regulators to propose adding session-specific quality-of-service controls to the Internet. The perceived need for lawful intercept leads to regulations that mandate that Internet services be "tappable." The perceived need to protect the incumbent telephone carriers' business viability produces regulations that permit those carriers to offer only Internet services in which service providers such as Google must pay to reach customers, and VoIP service providers that directly compete with services provided by the carrier could find it very hard to work out deals in which their traffic is not interfered with at prices they can afford while still being able to compete.

End-to-end security is not dead yet, but it is seriously threatened, at least at the network layer. NATs and firewalls interfere with some types of end-to-end encryption technology. ISPs could soon be required by regulations to, by default, filter the Web sites and perhaps the protocols that their customers can access. Other ISPs want to be able to limit the protocols that their customers can access so that the ISP can give service providers an "incentive" to pay for the customer's use of their lines—they don't see a way to pay for the network without this ability. The FBI has asked that it be able to review all new Internet services for tapability before they're deployed, and the FCC has hinted that it will support the request (see http://hraunfoss.

fcc.gov/edocs_public/attachmatch/FCC-05-151A1.pdf). If this were to happen, applications such as Skype that use end-to-end encryption could be outlawed as inconsistent with law enforcement needs.

Today, it's still easy to use end-to-end encryption as long as it's HTTPS, but that might be short-lived. It could soon reach the point that the use of end-to-end encryption, without which end-to-end security can't exist, will be seen as "an antisocial act" (as a US justice department official once told me). If that comes to be the case, end-to-end security will be truly dead, and we will all have to trust functions in the network that we have no way of knowing are on our side. □

References

1. J. Saltzer, D. Reed, and D. Clark, "End-to-End Arguments in System Design," *ACM Trans. Computer Systems*, vol. 2, no. 4, 1984; http://Web.mit.edu/Saltzer/www/publications/endtoend/endtoend.txt.
2. D. Isenberg, "Rise of the Stupid Network," *Computer Telephony*, Aug. 1997, pp. 16–26; www.isen.com/stupid.html.
3. D. Clark, "The Design Philosophy of the DARPA Internet Protocols," *Proc. SIGCOMM '88, Computer Communication Rev.* vol. 18, no. 4, Aug. 1988, pp. 106–114; www.acm.org/sigs/sigcomm/ccr/archive/1995/jan95/ccr-9501-clark.pdf.
4. A. Greenspan, "The Challenge of Central Banking in a Democratic Society," remarks at the Ann. Dinner and Francis Boyer Lecture of the American Enterprise Inst. for Public Policy Research, 5 Dec. 1996; www.federalreserve.gov/boarddocs/speeches/19961205.htm.
5. M. Gaynor and S. Bradner, *Firewall Enhancement Protocol (FEP)*, IETF RFC 3093, 1 Apr. 2001; www.ietf.org/rfc/rfc3093.txt.

***Scott Bradner** is the university technology security officer at Harvard University. His research interests include advanced network technologies and network security. Bradner is coeditor of* IPng: Internet Protocol Next Generation *(Addison-Wesley, 1996). He is a member of the IEEE, the American Bar Association, the Internet Society, and the ACM. He writes a weekly column for* Network World. *He sits on the* IEEE Internet Computing *editorial board. Contact him at sob@harvard.edu.*