Everything VPN is *New* Again

DAVID CRAWSHAW

THE 24-YEAR-OLD SECURITY MODEL HAS FOUND A SECOND WIND.

The VPN (virtual private network) is 24 years old. The concept—cryptographically secure tunnels used as virtual wires for networking—was created for a radically different Internet from the one we know today. As the Internet grew and changed, so did VPN users and applications. The VPN had an awkward adolescence in the Internet of the 2000s, interacting poorly with other widely popular abstractions such as multiuser operating systems. In the past decade the Internet has changed again, and this new Internet offers new uses for VPNs. The development of a radically new protocol, WireGuard, provides a technology on which to build these new VPNs.

This article is a narrative history of the VPN. All narratives necessarily generalize and cannot capture every nuance, but it is a good-faith effort to (critically) celebrate some of the recent technical history of networking and to capture the mood and attitudes of software engineers and network administrators toward the VPN.

THE FIRST AGE: FIEFDOMS AND LEASED LINES

Before the Internet there were networks: corporate networks, university networks, government networks. These networks were made of relatively expensive computers, had relatively few trusted people (at least by the standards of today's multibillion-person Internet), were managed by full-time network administrators, and were geographically clustered into buildings or campuses.

When an organization was split across more than one site it connected its networks with a leased line. In the 1970s this was dedicated unswitched copper wiring provided by a phone network to run a proprietary protocol such as DECnet.[3] Leasing a physical wire across hundreds or thousands of miles was not cheap. As phone networks became more sophisticated the leased line evolved into frame relays and connections to an ATM (asynchronous transfer mode) network. These networks reduced the cost of leased lines from the astronomical prices affordable by only the largest enterprises to merely very high prices accessible to a few more big companies.

The security model was physical and contractual. On-site networks were kept safe because the wall jacks into the network were guarded by guards and access badges. The leased lines were similarly guarded, so went the theory, by the phone company. Large contracts certainly felt safe.

Through the 1980s and 1990s the Internet was busy being built, under many names and in various places, by organizations interconnecting their networks using leased lines and by ISPs (Internet service providers) offering relatively cheap dial-up access to one of these peering networks. Many smaller organizations spread across

multiple sites could not afford a leased line but could afford to have each of their sites connected to a local ISP.

This raised an interesting possibility: Could a smaller organization get the benefit of a leased line connecting its sites over the internet? The VPN was born.

Could a smaller organization get the benefit of a leased line connecting its sites over the internet? The VPN was born.

PPTP

Several projects in the early 1990s worked on IP-layer security. The first one that could be called a VPN was swIPe.[6] The draft standard dates it to 1993. In swIPe, IP datagrams are encapsulated for encryption and then transmitted over another network, so you can make the claim that this is a VPN. It was never widely deployed.

The first unambiguous VPN was PPTP (Point-to-Point Tunneling Protocol). It was created in 1996 and standardized in RFC 2637 in 1999.[5]

Fittingly, PPTP was the product of a company that, at the time, produced networking software used by smaller businesses: Microsoft. (Microsoft did not have a stellar reputation as a network company in the 1990s, though today it runs, in terms of revenue, the second-largest cloud provider.)

PPTP worked, in its way. It played a great game of pretend, packaging up PPP (Point-to-Point Protocol), encrypting the stream, and running it over a TCP (Transmission Control Protocol) socket. Working as a virtual PPP, PPTP was able to encapsulate several network protocols, including an alternative to TCP/IP that was popular at the time among smaller organizations: IPX (Internet Packet Exchange), the protocol used by Novell's NetWare.

The cryptographic algorithms employed by PPTP—RC4 (Rivest Cipher 4) and DES (Data Encryption Standard)—are long obsolete, but even when these algorithms were considered adequate, several flaws in PPTP's implementation created security vulnerabilities. Such vulnerabilities would become an ongoing thorn in the side of the VPN.

IPSec

The 1990s were hectic. In the miasma of 1993 (top chart song that year: "Achy Breaky Heart"), IETF (Internet Engineering Task Force), an open-standards organization fresh from victory standardizing TCP/IP version 4, formed an IPSec (IP security) committee. IETF's goal was to bring security to IP. This was a much broader focus than creating what we know of today as a VPN, and the result was a standard that does a lot of things.

The first-draft RFCs (requests for comments) from the IETF for an all-purpose secure IP encapsulation standard were published in 1995. A working prototype came after PPTP, and the specification was published in RFC 2401 in 1998[7] with implementations starting to appear shortly thereafter.

A historical aside: the first-draft specifications of IPSec predate PPTP, and the first prototypes postdate PPTP, which raises the question of whether there was any connection between these two projects. It seems unlikely; instead this appears to be a case of history happening all at once. Indeed, two other VPN protocols were developed between PPTP's creation in 1996 and standardization in 1999. L2F (Layer 2 Forwarding) was a Cisco PPP-over-IP

protocol developed in 1997 and standardized in RFC 2341.[11] L2TP (Layer 2 Tunneling Protocol) is another protocol that borrows from both L2F and PPTP, though it was not standardized until later. Even more confusingly, the first RFC to use the term *VPN* is RFC 2194, published in 1997.[1] At this point the VPN had existed for only a year, but this RFC mentions three different protocols: PPTP, L2F, and L2TP (but not IPSec).

IPSec does everything, and everything it does is configurable. It has two modes of operation—tunnel and transport—multiple cryptographic suites, and multiple independent implementations. The result is an all-purpose toolkit for constructing networks that is still in use today. Like all things that try to be everything to everyone, however, for many of us the prospect of setting it up (and, more importantly, maintaining it) is daunting.

THE SECOND AGE: SATELLITE OFFICES AND CONSUMER PRIVACY

By the early 2000s in the United States it was possible for almost all desktops and laptops to reach the Internet, though many remained disconnected for reasons of policy, price sensitivity, or lack of adequate networking software. (Often this lack of software meant not that it was impossible to route a local network onto the Internet, but that doing so required a great deal of manual intervention by an expert, and experts were in short supply.)

More people wanted to connect offices together; small businesses wanted their employees to reach their exchange servers from satellite offices; the early (and at the time rare) remote workers wanted the same

experience at home or on the road that they got in the office. The VPN had to become easier to configure and integrate with a user-friendly authentication scheme.

SSL/TLS

In a separate development, web browsers had developed a robust means of encrypting traffic: SSL (Secure Sockets Layer), later TLS (Transport Layer Security), which involved distributing a trusted set of root certificates to clients used to identify servers. With web browsers finding their way onto all computers, reusing these certificates for VPNs provided an easy way to deploy VPN gateways that remote users and satellite offices could connect to in a hub-and-spoke arrangement.

One notable open-source VPN product built on these principles is OpenVPN, which lets users authenticate with a username/password and connect to a VPN gateway that is authenticated with a TLS certificate. This project is active and forms the foundation of many corporate and consumer VPNs today.

Consumer VPNs

As more consumer activity moved onto the Internet, traditional businesses that relied on restricting consumer access across geographies to maximize revenue began to introduce services locked to regions, mechanically enforced by looking at the user's IP address. Simultaneously, a new industry of targeted advertising was developed that tracks users by using their IP addresses to determine interests and spending habits.

In response to these industries, consumer VPN products

were developed and became moderately popular. These are very different from the corporate VPN, whose objective is to move a packet between two trusted endpoints without revealing it to intermediate carriers. The consumer VPN, on the other hand, hides traffic from the consumer to the VPN gateway, and then as a proxy for the consumer sends the packet to a public server. This hides the consumer's IP address, confounding these new industries.

This new product introduces an interesting terminology challenge when talking about VPNs. The technology is the same—an encapsulated IP tunnel—but the application is radically different. A consumer VPN ensures "privacy" for only the part of the traffic transferred between the consumer device and the VPN gateway; after that it is public again. But the consumer VPN is widespread and useful, so the industry now refers to both products as VPNs.

Limitations: identity

The VPN does a decent job of encrypting traffic over public networks, but one weakness of the model traditionally is that IP addresses do not line up with authorization identities. There are two variants of this problem.

First, multiple users on a single computer all share an IP address. Network stacks are traditionally considered a computerwide service provided by the operating-system kernel. Thus, any user on a computer can act as a VPN tunnel's IP address.

Second, if a VPN gateway is used to connect two subnets of machines together, there is no way to map the credentials used to establish the VPN to identify the machines on the subnet it routes. You have expanded your

network to include all of the network on the other end of the VPN. If you don't manage both networks yourself, then you have just created a new network administrator.

These limitations are frustrating because they mean a traditional corporate VPN does not provide user security. All the effort put into managing credentials on both ends of the VPN has to be done again one layer up, between users over the VPN.

Disillusionment: BeyondCorp and Zero Trust

The scale of the Internet strains the security model of the second-age VPN. Smaller organizations using the VPN as a cheaper leased line, then letting their traveling workers use that leased line from any hotel room, face a growing problem: With far fewer resources than a large organization, the small organization needs to secure an access point to its network that is getting ever cheaper to attack. As the number of Internet users went from millions to billions, corporate VPN gateways went from thousands to (potentially) millions. Each of these gateways runs one of a handful of implementations, uses common usernames, lacks two-factor authentication, and has common unmanaged passwords. An attack written to find an exploit on some fraction of existing targets becomes more cost effective as the number of targets grows.

The growing threat interacts poorly with the "eggshell" security model (hard exterior, soft gooey interior) of traditional corporate VPNs: the idea that the VPN keeps your network safe, so you can be lax about what you transmit. As VPNs become more cost-effective targets as their numbers grow, corporate VPNs get compromised

more often.

This has led several security experts to call for VPNs to be dismantled.[10] New security models have been proposed to replace the VPN. Two notable approaches are BeyondCorp,[12] a project by Google to secure its corporate network infrastructure; and a developing industry idea known as the *zero trust network*.

BeyondCorp and zero trust have a lot of conceptual overlap and can best be summarized as applications of the venerable end-to-end principle from computer networking. Specifically, when any two services communicate, each service must mutually authenticate who it is talking to and ensure that the other service is authorized to communicate.

This concept of mutual authentication is incompatible with the traditional corporate VPN "gateway" or "concentrator," a device that sits on a network and routes all traffic through it, because the devices on either side of the gateway cannot be sure the other devices are who they say they are.

The second age of the corporate VPN is coming to an end. Its security model is incompatible with modern Internet scale. It is too soon to declare the death of the VPN, however. Consumer VPNs continue to be useful tools, and while the traditional way of configuring corporate VPNs is clearly over, the underlying concept of encapsulating an IP packet still works. Put another way, the corporate application needs rethinking, but there still may be a use for the technology.

THE THIRD AGE: SINGLE-USE DEVICES AND VIRTUAL NETWORK NAMESPACES

The big exciting VPN development of the past few years is WireGuard,[4] a completely new implementation of IP encapsulation using the latest in cryptographic algorithms and principles.

WireGuard

WireGuard, the creation of Jason A. Donenfeld, is built on top of the cryptographic primitives curve25519 and chacha20.[13] The protocol creates a tunnel between two equal peers, each identified with public/private key pairs rather than the common client-server architecture of VPNs with gateways and concentrators. It adopts handshake techniques and principles of the Noise Protocol[9] to make it practically impossible for adversaries even to know a machine is running a WireGuard endpoint. There is no standard port to scan for on a network.

What makes WireGuard radical, however, is not its adoption of the very latest in cryptographic algorithms (which many would consider a classic virtue of developing a product from scratch). WireGuard is radically simple. It has only one cryptographic suite. There is no version-negotiation phase of the protocol, where multiple implementations try to agree on how to talk. It tries to be exactly one thing: a secure encrypted tunnel between two endpoints.

In a world where networking software tries to be everything to everyone, where configuration languages need their own independent standards committee,[2] WireGuard is a breath of fresh air. It simplifies encrypted tunnels to the point where you can stop thinking of it as the final product, and

instead consider it a fundamental abstraction around which software and networks can be designed.

Single-use devices and Zero-trust VPNs

In addition to an exciting new VPN protocol, there have been fundamental shifts in the way computers are used that create new uses for VPNs and solve old security issues.

The most significant technical shift in network computing of the past decade has been the rise of the single-use device. This is driven by two changes to computing.

First, interactive user devices are effectively single-user operating systems because they allow only a single user at a time. The most extreme example of this is iOS—more than a billion active devices—which does not even support multiple user accounts on a single device. But even more traditional desktops and laptops typically have only one logged-in user. Rarer devices that support fast user switching can be configured so the VPN disconnects as one user and connects as another on a switch. Large shared Unix minicomputers with terminals are now interesting hobby projects, not typical corporate setups.

The second change is the near-universal virtualization of servers with virtual machines or container technology such as Linux namespaces. There are several forces driving this change and several ways it is achieved, but the result is the same: A server ends up running on a multiuser, multitasking Unix operating system with effectively a single-purpose process and one user.

In both of these cases, the result is that the operating system's virtual tunnel IP addresses now line up with service identities used for authorization. On end-user devices an IP

address is a user, and in data centers each service instance has its own IP. With WireGuard ensuring every packet with a particular IP source is cryptographically linked to a verifiable identity, we can start safely making statements such as, "Address *a* is user *u*," which simplifies software development. Tailscale[8] is an implementation of a VPN-identified network built on WireGuard.

The growing number of end-user devices and a new layer of virtualization in data centers has subtly but profoundly changed how the VPN abstraction fits into networking. With a little care in a modern environment, the traditionally awkward and unhelpful security model of the VPN suddenly fits perfectly and solves problems instead of creating them. This is what makes the third age of the VPN so exciting: The clumsy '90s child, a millennial often dismissed as awkward and out of place, is suddenly making computing easier and better.

References

1. Aboba, B., et al. 1997. Review of roaming implementation. IETF Network Working Group; https://tools.ietf.org/html/rfc2194.
2. Ben-Kiki, O., Evans, C., döt Net, I. 2009. YAML specification index; https://yaml.org/spec/.
3. Digital Equipment Corporation. 1978. Digital Equipment Corporation: Nineteen fifty-seven to the present, p. 53. Computer History Museum; https://www.computerhistory.org/pdp-1/8a9cb4c9f949fbb3e577016d174499ca/.
4. Finley, K. 2020. WireGuard gives Linux a faster, more secure VPN. *Wired* (March 2); https://www.wired.com/

story/wireguard-gives-linux-faster-secure-vpn/.
5. Hamzeh, K., et al. 1999. Point-to-Point Tunneling Protocol. IETF Network Working Group; https://tools.ietf.org/html/rfc2637.
6. Ioannidis, J., Blaze, M. 1993. The swIPe security protocol. Internet draft; https://www.mattblaze.org/papers/swipe.id.txt.
7. Kent, S., Atkinson, R. 1998. Security architecture for the Internet Protocol. IETF Network Working Group; https://tools.ietf.org/html/rfc2401.
8. Pennarun, A. 2020. How Tailscale works. Tailscale; https://tailscale.com/blog/how-tailscale-works/.
9. Perrin, T. 2018. The Noise Protocol Framework; http://noiseprotocol.org/noise.pdf.
10. Sullivan, P. 2019. The death of the VPN—it's time to say goodbye. *SC Magazine* (March 21); https://www.scmagazine.com/home/opinion/the-death-of-the-vpn-its-time-to-say-goodbye/.
11. Valencia, A., et al. 1998. Cisco Layer 2 Forwarding Protocol. IETF Network Working Group; https://tools.ietf.org/html/rfc2341.
12. Ward, R., Beyer, B. 2014. BeyondCorp: a new approach to enterprise security. *;login;* 39(6), 6-11; https://research.google/pubs/pub43231/.
13. WireGuard; https://www.wireguard.com/protocol/.

David Crawshaw *is cofounder and CTO of Tailscale. Before that, he worked on a variety of software projects, including the Go programming language.*