Virtual Private Networks

The Problem

We want to use remote computers: branch offices, telecommuters, travelers, etc. Is that a secure thing to do?
Assumptions:

1. The Internet is a bad place
2. Firewalls protect us from those bad things, so we want to keep all of our computers inside our firewall
3. (Optional) The bad people are tapping our links, too

We need a network that is secure nevertheless

Real Links

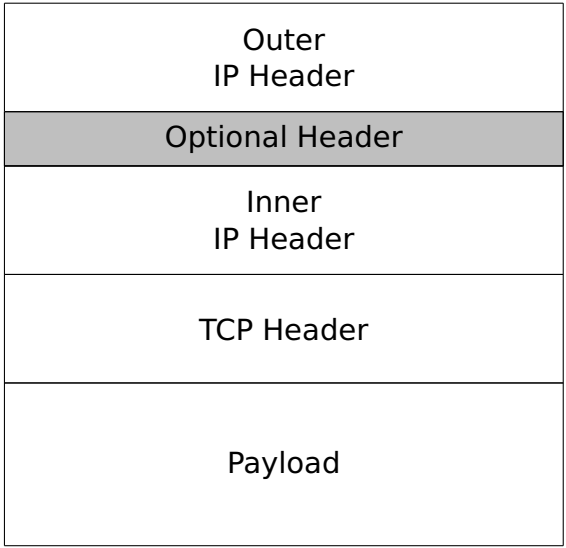
- Maybe we should lease lines from the phone company
- That's expensive and inflexible for branch offices, and doesn't help with telecommuters, let alone road warriors
- Besides: do we trust the phone company?

Solution: Virtual Private Networks (VPNs)

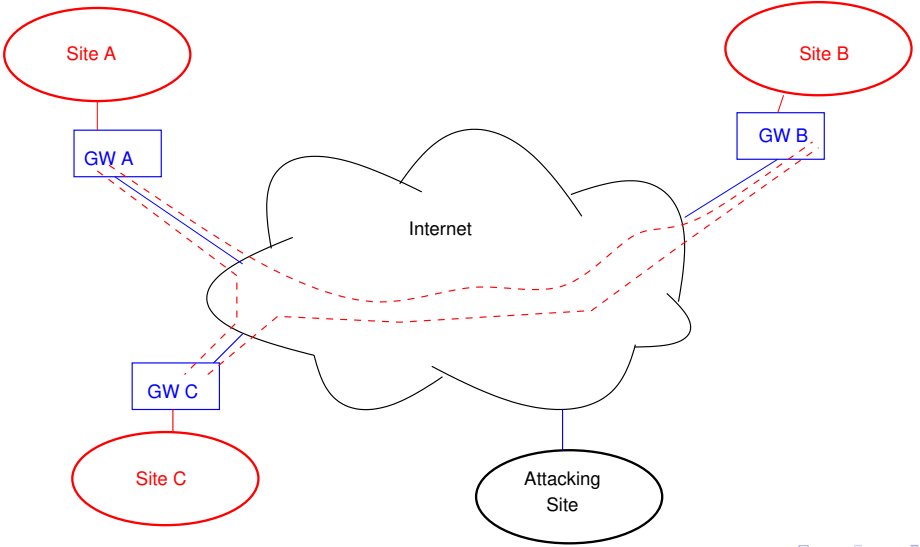
- Send the data over the Internet itself
- However—wrap the data (somehow!) to keep outside attackers from getting in
- Optional: encrypt the data

Tunneling

- Encapsulate IP packets in an outer IP header
- Optional extra header
- The outer IP header gets the packet from gateway to gateway
- The inner IP header is used inside the networks behind each gateway
- This is called *tunneling*



Tunneling



IP in IP

- Simplest form: no optional header
- Have the Next Protocol field in the outer IP header be set to 4: IPv4

IP in IP

- Simplest form: no optional header
- Have the Next Protocol field in the outer IP header be set to 4: IPv4
- What's wrong?

Gateway Operation

Outbound Packets from GW A

1. Look at destination IP address of packet
2. Locate proper outbound gateway, e.g., GW-C
3. Construct outer IP header: src=GW-A,dst=GW-C
4. Send packet

Inbound Packets from GW-A

1. Verify legal gateway address, e.g., GW-A
2. See if inner IP source address belongs to GW-A
3. See if inner IP destination address belongs here
4. Forward packet internally

Gateway Operation

Outbound Packets from GW A

1. Look at destination IP address of packet
2. Locate proper outbound gateway, e.g., GW-C
3. Construct outer IP header: src=GW-A,dst=GW-C
4. Send packet

Inbound Packets from GW-A

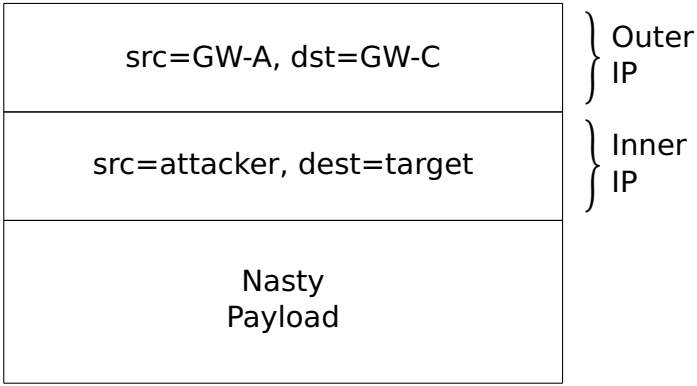
1. Verify legal gateway address, e.g., GW-A
2. See if inner IP source address belongs to GW-A
3. See if inner IP destination address belongs here
4. Forward packet internally

What's wrong?

IP Address Spoofing!

- As we've discussed, it's easy to spoof IP addresses
- The attacking site can send bogus IP-in-IP packets to a gateway and inject packets into a target network
- (Both the inner and outer source IP addresses are spoofed)
- Return packets won't go back to the attacker—can you successfully attack that way? Sometimes!
- (What would happen if the inbound gateway didn't verify the plausibility of the inner source IP address?)

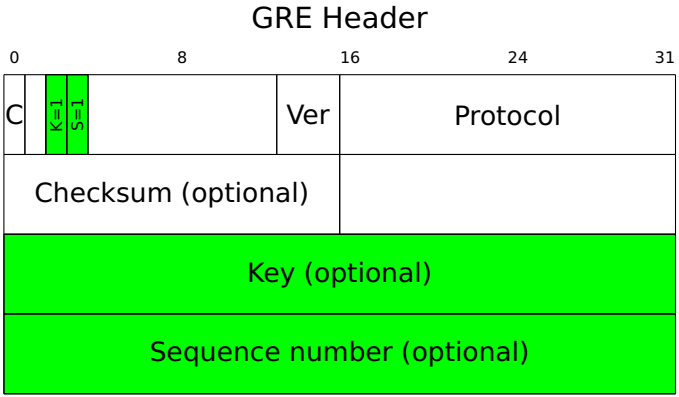
Spoofing the Inner Source Address



The reply packets from the target will go back to the attacker, so there can be a complete TCP connection set up!

Authenticated Tunnels

- Gateways need to be able to authenticate inbound packets
- Simplest solution: a plaintext “key”—really, a shared secret; it’s not used for encryption—on every packet
- This is commonly done for Generic Router Encapsulation (GRE)

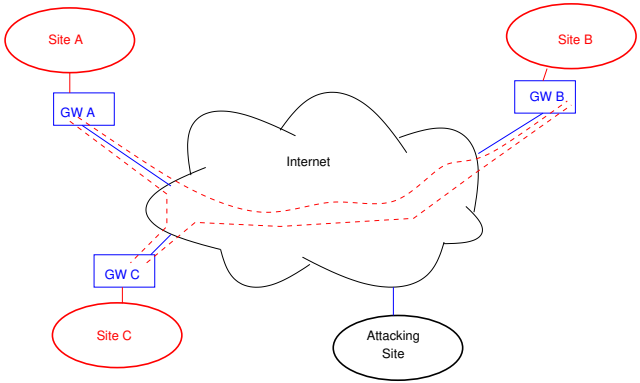


That’s Not Enough

- How do we coordinate “key” changes?
- Is a 32-bit key secure enough?
- How do we provision proper gateway knowledge?
- What if we don’t trust the ISPs?
- We need cryptography and more

Encrypting VPNs

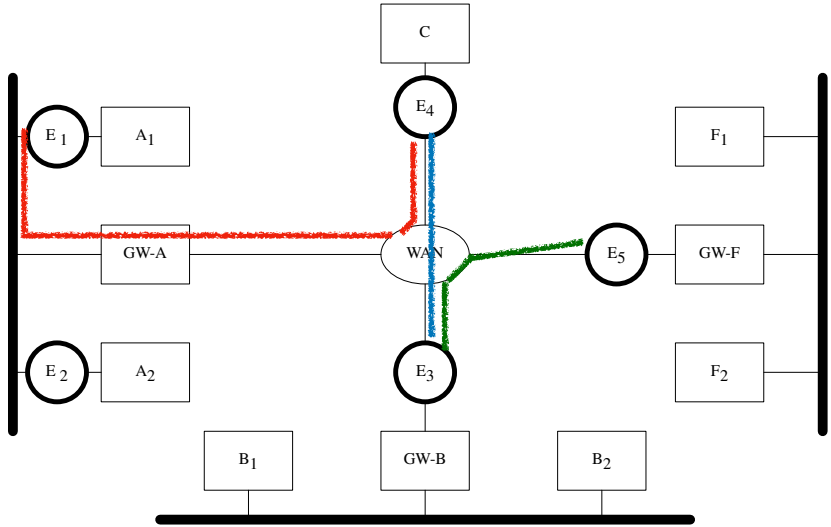
- Suppose that GW-A, GW-B, etc., encrypt and decrypt packets
- Packet-forging becomes impossible
- We no longer need to trust the provider except for availability
- ☞ This is what most people mean by VPNs
- But what do the packets look like?



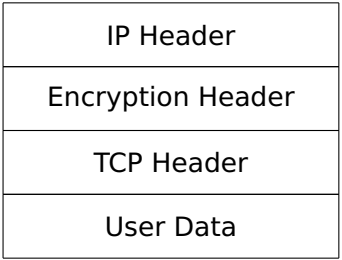
IPsec: The IETF's VPN Standard

- Based on earlier DoD and research efforts
- Supported host-to-host, host-to-gateway, gateway-to-gateway
- Separate over-the-wire protocol from key management
- ☞ Separate policy from mechanism
- Went through a few iterations before we got it right
- Still a few missing pieces

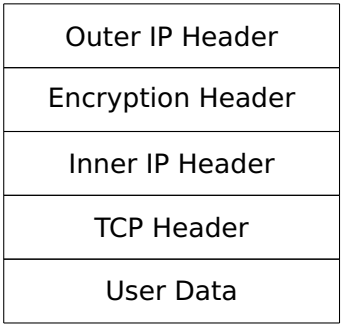
Topologies



Overall Structure

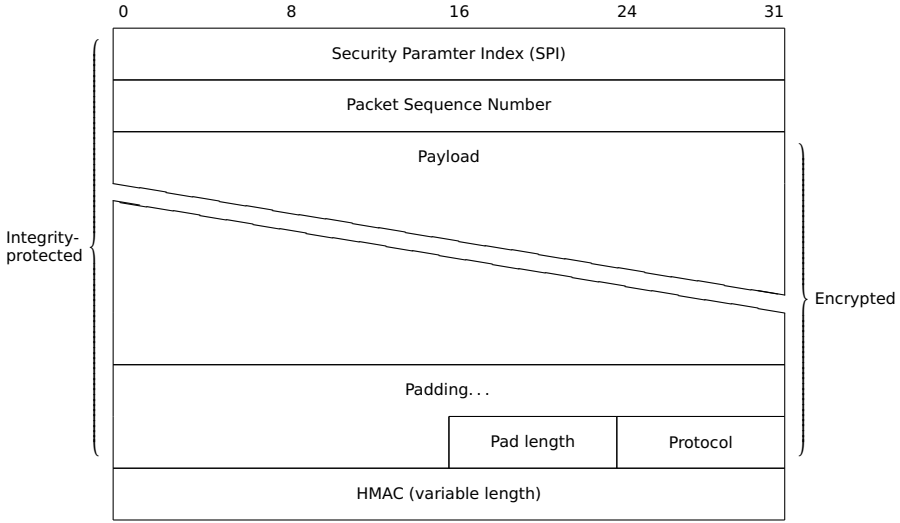


End system to end system



End system to gateway
or gateway to gateway

ESP: Encapsulating Security Protocol



The ESP Header

SPI	Index to encryption and security parameters:
Sequence Number	Many attacks possible if this is omitted!
Pad Length	Accommodate block cipher blocksize
Protocol	Identity of next protocol header: IP, TCP, etc.

The Security Parameter Index

- Points to many things: encryption algorithm, encryption blocksize, integrity algorithm, integrity field length, IP address range, etc.
- Keeps things like algorithm identifier out of the packet—shortens the packet, and the knowledge may help the attacker
- Address ranges: what are the legal IP addresses for this SPI?

IPsec Operation: Outbound

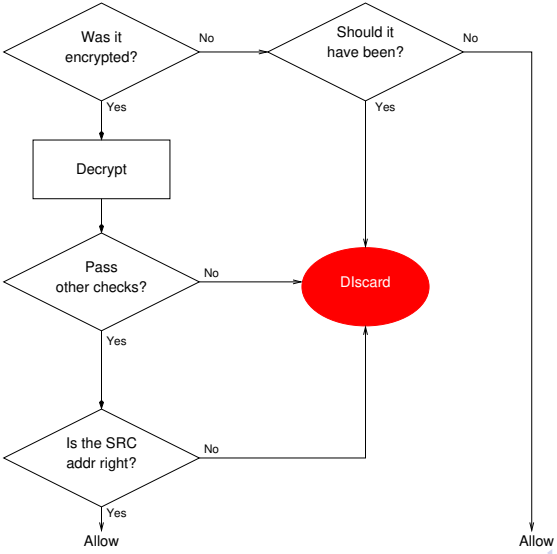
1. Consult the *Security Policy Database* (SPD) for, e.g., destination IP address of this packet
2. Should this packet be encrypted? If not, just forward it
3. If so: is there a *security association*? If not, negotiate one
4. If there is an association: encrypt the packet according to the negotiated parameter

IPsec Operation: Inbound

1. Was this packet encrypted? Per the SPD, should it have been?
2. If it should have been encrypted but isn't, drop it; if it shouldn't have been, pass it through
3. If it was encrypted, decrypt it (and perform other checks, e.g., sequence number and integrity)
4. Does the source IP address match what's legal for this security association? If not, drop the packet

☞ As with simple tunneling, must guard against malicious packet injection

Inbound Processing



Negotiating Security Associations

- Execute a cryptographic protocol between the two security endpoints
- (IPsec gateway discovery still doesn't exist—hard to do securely)
- *Many* variations (and very complex)
- Multiple forms of authentication supported: passwords, key pairs, tokens, etc.
- Done rarely: move out of mainline processing, do at user level instead of in the kernel, etc.
- (A long, complex, crazy story, involving personalities, organizational politics, corporate interests, the NSA, and more. . .)

What Did IPsec Attempt?

- Ubiquitous encryption
- Protect all host-pairs
- Protect all traffic, for all applications
- We didn’t get it. . .

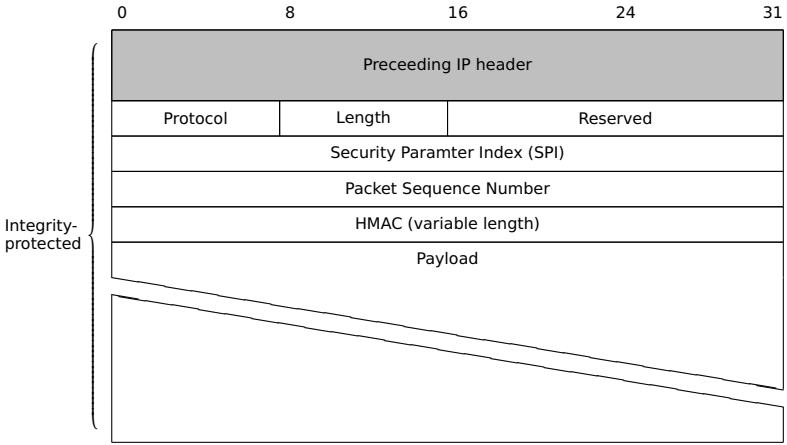
Why Not?

- Too much complexity in security association negotiation
- The design took too long—the first version had fatal flaws and had to be discarded
- Purists on the design team rejected network address translation (NAT)
- End-to-end encryption didn’t play well with firewalls
- Computers then were too slow to encrypt everything
- US export restrictions

Authentication Only

- The last three issues—firewalls, speed, and export controls—were tied to confidentiality
- (The only accepted encryption algorithms were DES and 3DES, which are slow in software)
- Result: a design for a different IPsec protocol, AH: Authentication Header
- AH provided integrity only
- But: it still required a security association

AH: Authentication Header



AH is Problematic

- It violates layering—you have to authenticate portions of a lower-layer protocol
- Parts of the IP header can change en route
- We can do ESP in most situations

Complexity is Fatal

- Because of the complexity of the security association negotiation, different implementations didn't interoperate very well
- Effectively, each became a proprietary solution
- Other—and simpler-to-configure—VPN technologies took over
- Example: OpenVPN, Microsoft's PPTP, Wireguard, and a variety of TLS-based setups

Provider-Provisioned VPNs

- Companies decided that they could trust their ISPs
- ISPs began offering variety of network-based VPNs
- They aren't encrypted—but they send traffic where it should be and don't let others' traffic impersonate it
- (Details are out of scope for this course)
- This works for branch offices—but what about road warriors?

Do Road Warriors Need VPNs?

- In 1994, to read your email you had to log in remotely to some server
- POP and IMAP were little-used and not well-supported
- Of *course* we needed VPNs!
- Today: maybe you use cloud-resident email and connect to the corporate net via Microsoft's Remote Desktop Protocol (RDP)
- We have many encrypted application protocols

Do We Need Internal VPNs?

- More and more, the challenge isn't keep the attackers out entirely—too many get in no matter what
- We need to prevent *lateral movement* within an organization
- In other words, the firewall has failed—but we still have to stop the attacker
- Answers: internal firewalls and encryption—which is often VPNs

VPNs Today

- Heavily used by telecommuters (especially today!)
- Risks from buggy code and configuration issues
- But: there are issues

Fate-Sharing

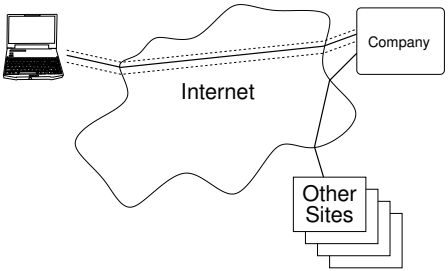
- Everyone exiting a firewall appears to have the same IP address
- This means that the behavior of any user of the VPN will be attributed to all of them
- In other words, VPNs can trigger false positives by intrusion detection systems

Firewall Configurations

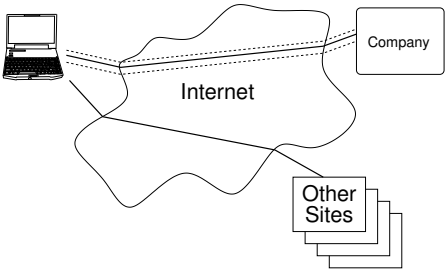
- Does your VPN send all traffic to the gateway, or does non-work traffic go direct to the Internet?
- Triangle routing: provide firewall protection for home laptops
- But—most web traffic is encrypted; does the firewall help?
- But—it's a lot of extra traffic; was your link bandwidth configured correctly for this scenario?

Triangle Routing versus Split Tunneling

Triangle Routing



Split Tunneling



Questions?

(Mandarin duck, Central Park, January 11, 2019)