Last time

- Internet Application Security and Privacy
  - Authentication
  - Security controls using cryptography
  - Link-layer security: WEP

This time

- Internet Application Security and Privacy
  - Link-layer security: WEP, WPA, WPA2
  - Network-layer security: VPN, IPSec

WEP data integrity

- Problem 2: the checksum used in WEP is CRC-32
  - Quite a poor choice; there's already a CRC in the protocol to detect random errors, and a CRC can't help you protect against malicious errors.
- The CRC has two important properties:
  - It is independent of *k* and *v*
  - It is linear: *c(M* XOR *D)* = *c(M)* XOR *c(D)*
- Why is linearity a pessimal property for your integrity mechanism to have when used in conjunction with a stream cipher?

WEP access control

- What if the adversary wants to inject a new message *F* onto a WEP-protected network?

- All he needs is a single plaintext/ciphertext pair
- This gives him a value of *v* and the corresponding keystream *RC4(v,k)*
- Then *C'* = <*F,c(F)*> XOR *RC4(v,k)*, and he transmits *v,C'*.
- *C'* is in fact a correct encryption of *F*, so the message must be accepted.

WEP authentication protocol

- How did we get that single plaintext/ciphertext pair we needed just now?
  - Problem 3: It turns out the authentication protocol gives it to the adversary for free!

- This is a major disaster in the design!
- The authentication protocol is supposed to prove that a certain client knows the shared secret *k*
- But if I watch you prove it, I can turn around and execute the protocol myself!
  - "What's the password?"

WEP authentication protocol

- Here's the protocol:
  - The access point sends a challenge string to the client
  - The client sends back the challenge, WEP-encrypted with the shared secret *k*
  - The base station checks if the challenge is correctly encrypted, and if so, accepts the client
- So the adversary has just seen both the plaintext and the ciphertext of the challenge
- Problem number 4: this is enough not only to inject packets (as in the previous attack), but also to execute the authentication protocol himself!

WEP decryption

- Somewhat surprisingly, the ability to modify and inject packets also leads to ways to adversary can decrypt packets!
  - The access point knows *k*; it turns out the adversary can trick it into decrypting the packet for him and telling him the result.

- Note that none of the attacks so far:
  - Used the fact that the stream cipher was RC4 specifically
  - Recovered *k*

Recovering a WEP key

- Since 2002, there have been a series of analyses of RC4 in particular
  - Problem number 5: it turns out that when RC4 is used with similar keys, the output keystream has a subtle weakness
    - And this is how WEP uses RC4!
- These observations have led to programs that can recover either a 104-bit or 40-bit WEP key in under 60 seconds, most of the time
  - See the optional reading for more information on this

Replacing WEP

- Wi-fi Protected Access (WPA) was rolled out as a short-term patch to WEP while formal standards for a replacement protocol (IEEE 802.11i, later called WPA2) were being developed
- WPA:
  - Replaces CRC-32 with a real MAC (here called a MIC to avoid confusion with a Media Access Control address)
  - IV is 48 bits
  - Key is changed frequently (TKIP)
  - Ability to use 802.11x authentication server
    - But maintains less-secure PSK (Pre-Shared Key) mode for home users
  - Able to run on most older WEP hardware

Replacing WEP

- The 802.11i standard was finalized in 2004, and the result (called WPA2) has been required for products calling themselves “Wi-fi” since 2006

- WPA2:
  - Replaces the RC4 and MIC algorithms in WPA with the CCMP algorithm, which uses AES
  - Considered strong, except in PSK mode
    - Dictionary attacks still possible

Network-layer security

- Suppose every link in our network had strong link-layer security
- Why would this not be enough?

- We need security across networks
  - Ideally, end-to-end

- At the network layer, this is usually accomplished with a Virtual Private Network (VPN)

Virtual Private Networks

- Connect two (or more) networks that are physically isolated, and make them appear to be a single network
  - Alternately: connect a single remote host (often a laptop) to one network
- Goal: adversary between the networks should not be able to read or modify the traffic flowing across the VPN
  - But DoS and some traffic analysis still usually possible

Setting up a VPN

- One host on each side is the VPN gateway
  - Could be the firewall itself, or could be in DMZ
  - In the laptop scenario, it will of course be the laptop itself on its side
- Traffic destined for the “other side” is sent to the local VPN gateway
- The local VPN gateway uses cryptography (encryption and integrity techniques) to send the traffic to the remote VPN gateway
  - Often by tunnelling
- The remote gateway decrypts the messages and sends them on to their appropriate destinations

Tunnelling

- Tunnelling is the sending of messages of one protocol inside (that is, as the payload of) messages of another protocol, out of their usual protocol nesting sequence
  - So TCP-over-IP is not tunnelling, since you're supposed to send TCP (a transport protocol) over IP (a network protocol; one layer down in the stack)
  - But IP-over-TCP is tunnelling (going up the stack instead of down), as are IP-over-IP (same place in the stack), and PPP (a link layer protocol; bottom of the stack) over DNS (an application layer protocol; top of the stack)

IPSec

- One standard way to set up a VPN is by using IPSec
- Many corporate VPNs use this (open) protocol
- Two modes:
  - Transport mode
    - Useful for connecting a single laptop to a home network
    - Only the contents of the original IP packet are encrypted and authenticated
  - Tunnel mode
    - Useful for connecting two networks
    - The contents and the header of the original IP packet are encrypted and authenticated; result is placed inside a new IP packet destined for the remote VPN gateway

Other styles of VPNs

- In addition to IPSec, there are a number of other standard ways to set up a VPN
- Microsoft's PPTP was an older protocol
  - It had about as many design flaws as WEP
  - Most users now migrating to IPSec
- VPNs based on ssh
  - Tunnel PPP over ssh
    - That is, IP-over-PPP-over-ssh-over-TCP-over-IP
    - Some efficiency concern, but extremely easy to set up on a standard Unix/Linux box
  - OpenSSH v4 supports IP-over-SSH tunnelling directly

Recap

- Internet Application Security and Privacy
  - Link-layer security: WEP, WPA, WPA2
  - Network-layer security: VPN, IPSec

Next time

- Internet Application Security and Privacy
  - Transport-layer security and privacy: TLS / SSL, Tor
    - The Nymity Slider
  - Application-layer security and privacy: ssh, remailers