Wireless Security

Wireless is Different

- Actually, why is wireless different?
- Is it different?

Wireless *is* Different

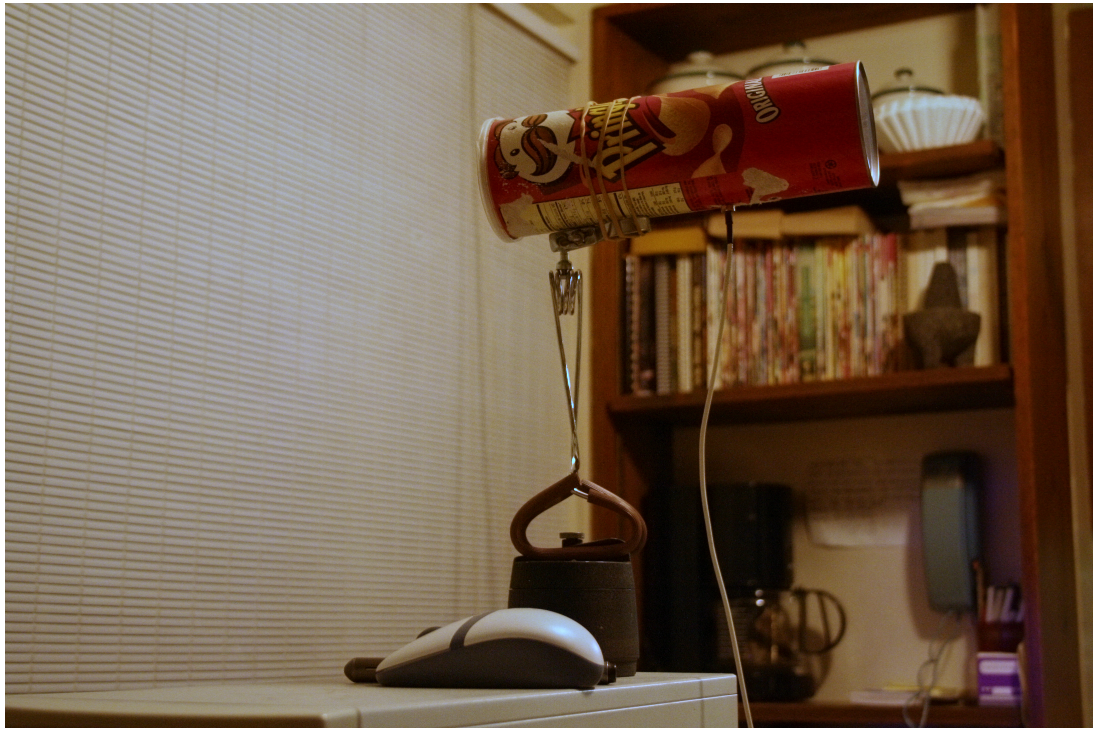
- The attacker has access to the network that isn’t as constrained by physical location
- Your security perimeter is much larger
- There are more protocols involved
- Traffic can be monitored
- Traffic can be injected

Common Types of Wireless

- Cellular—Range 1.5-2km from the nearest antenna
- WiFi—range of about 100 meters
- Bluetooth—nominal range of 10 meters
- NFC (Near-Field Communication)—4 cm range

Range

- All ranges can be limited by terrain, intervening objects, and more
- But use of proper antennas can extend the range—important to realize when analyzing security!
- Example: the classic Pringles Can WiFi antenna—perhaps 1 km range



Picture from https://www.flickr.com/photos/wmjr/106601670/ by WmJr.

(How Far Away is that Node?)

- If range limits matter, you *can't* trust the nominal limits
- Use the ultimate limit: the speed of light.
- Sets an upper bound on distance
- Example: a signal *cannot* do a round trip of 5 cm each way in less than .16 nanoseconds
  - (Grace Hopper used to hand out foot-long pieces of wire and describe them as "nanoseconds")
- (The actual calculation is a bit more complex, but it can only raise that limit)

Possible Attacker Goals

Host Access: Ability to talk to a given computer on the net

Network Access: Ability to act as a legitimate computer on the wireless net

Content Access: Ability to read packets sent and received

Metadata Access: Ability to conduct traffic analysis

NFC

- Primarily used for payments
  - The same basic protocol is used for RFID credit cards
- Relatively low bandwidth—106–424K bps
- Often used by very low power devices, e.g., RFID chips—hard to do much crypto
- Sometimes used to set up faster connections, e.g., between Android phones

NFC: Attacker Goals

- Content access, especially for payment card info
- Better yet, spoof a payment for something else
- Possibility for some host access for phone-to-phone NFC
  - Supported on some Android versions

NFC—Attacks

- Hard; none known in the wild
- Timing matters for NFC exchanges
- Some MitM and relay attacks have been demonstrated in the lab
- More serious issue: bugs in the NFC protocol stack
- *Complexity leads to insecurity!*

Bluetooth

- Used for short range, moderate bandwidth communications
- Typical uses: wireless keyboards and mice, headphones, body-area networks, bootstrapping faster communications, etc.
- Bandwidth: up to 1.4 Mbps for Bluetooth 5.0; slower for earlier versions
- Security issue: *pairing*
- Pairing: how does one Bluetooth device know which other device to connect to and to encrypt to?

Bluetooth Pairing

- Many variants, depending on device type
- Fancier devices can require PIN entry
- Simpler devices, e.g., headphones, might enter pairing mode when you do something odd such as holding down the power button
- A pairwise secret is negotiated; this is used for future associations and communications after successful pairing

Bluetooth Attacks

- Many…

Bluetooth Attacks

- Cryptographic flaws
- Protocol flaws
  - The Bluetooth protocol stack is *very* complex
- Pairing flaws
- Implementation flaws

In other words, more or less anything that can go wrong, has…

WiFi

- Extremely common
- Intended as “wireless Ethernet”—replacement for traditional, high-speed, wired networks
- Speed varies with version, range, and WiFi usage in the area, but generally in the 10s of megabits/second.
- Today, used far more than originally anticipated, for phones, tablets, IoT, and more

WiFi Encryption

- WiFi had encryption from the very beginning—but it had a long, tortuous history
- The goal of the original standard, WEP, is told in its name: “Wired-Equivalent Privacy”
- In other words: make the security equal to that of wired Ethernet, no less—but no more
- It didn’t succeed…

WEP

- Shared key among all users of a network
- No key management—keys were static
- Used RC4, a stream cipher, and an unkeyed CRC instead of a MAC
- Originally used a 40-bit key, due to US export rules; later raised to 104 bits
- Why RC4? Remember the limitations of 1999 hardware—anything better was deemed too expensive, in silicon and in battery power. RC4 is *very* efficient
- But: WEP was a horrible failure as a security mechanism

MAC versus MAC

- MAC: Message Authentication Code, a keyed cryptographic checksum that detects unauthorized modifications to the message
- MAC address: Media Access Control address, i.e., an Ethernet address

WiFi: A Packet Medium

- WiFi—like Ethernet and for that matter IP—is packet-oriented
- Each packet is an independent message
  - Each packet is self-contained
  - Packets may be dropped, duplicated, damaged, or reordered
  - Any stream-like semantics have to be handled at a higher protocol layer (on the Internet, that's TCP)
- This means that encryption *at the WiFi layer* has to be packet-oriented

WiFi and WEP Encryption

- Ideally, WiFi would use a block cipher in some suitable mode of operation, e.g., CBC
  - (Modes like GCM hadn't been invented yet)
- Stream ciphers assume a reliable underlying byte sequence—but on a packet network, there is no reliable layer larger than a packet
- WEP used RC4, a stream cipher, so every packet had to be encrypted independently
- RC4 generates a pseudo-random byte stream that is XORed with the plaintext, a byte at a time

  $$C_i := P_i \oplus S_i$$

  but for WEP, this can only be done within a packet

WiFi and Stream Ciphers

- With stream ciphers, it is *vital* not to reuse a key stream for two different plaintexts

$$C_{1,i} \oplus C_{2,i} = P_{1,i} \oplus S_i \oplus P_{2,i} \oplus S_i = P_{1,i} \oplus P_{2,i}$$

- If you know one byte, it gives you the other, or you can guess at one plaintext stream and see if it makes the other make sense
- To avoid this, WEP used a 24-bit IV that was concatenated with the static key to form a longer effective key
- 24 bits wasn't nearly enough, and they didn't even specify it properly

The IV Problem

- 2^{24} isn't very many packets—an access point will send that many fairly quickly
- The standard didn't say how IVs were to be selected—and many devices always started at 0 on power-up (which was frequent, since early WiFi cards were removable from laptops)
- If an implementation tried to be smart and use random IVs, instead of sequential ones, it would repeat on average every ~5,000 packets (birthday paradox)
- The spec didn't even bar repeated use of the same IV!

It's worse!

Packet Injection

- Suppose you know the full content of a packet
  - How? Send the target machine such a packet
  - Ping the target, and get the ping and the reply
  - Or induce the target to send you email or visit your website
- XOR the known packet against the ciphertext of a WEP-protected WiFi packet
- That gives you the S_i for the entire packet—use that to create as many new packets as you wish
- Many more variations on this game

But it got worse!

RC4 Isn't Very Good

- RC4 turned out not to be very strong against cryptanalysis
- Especially in the context of WEP, it's easy to crack
- Result: breaking into WEP-protected networks is more or less the only widespread use of a cryptanalytic attack in the wild

WEP Operational Issues

- The lack of key management meant that there were no session keys—recovering the WEP key was all you needed for full access
- Since everyone in an organization shared the same key, changing it was logistically almost impossible; everyone had to do it at the same time or they'd lose access

What Went Wrong with WEP?

- The hardware was underpowered
- The designers of WEP knew too little about cryptography—using a stream cipher was simply wrong
  - They should have tried to find a low-energy block cipher
  - At the very least, they could have used a much longer IV; it would have helped against many of the problems, with almost no performance hit
  - They should have used a keyed checksum
- They left key management to a higher level of the protocol stack, but it was never designed, let alone implemented or adopted
- There were no knowledgeable eyes on the entire standardization process

Consequences

- WEP attacks have been used in the real world
- The TJX attack is just one example

WEP versus our Threat List

Host Access: Available due to weak checksum and use of a stream cipher

Network Access: Available via the known full packet attack

Content Access: Available due to cryptanalytic weakness

Metadata Access: The source and destination MAC addresses are sent in the clear

WPA2: WiFi Protected Access

- *Much* stronger than WEP
- Uses AES and a real MAC
- Two modes: WPA2 Personal, with a single pre-shared key for the network, and WPA2 Enterprise, which has a login and password per user
- Still some cryptographic issues—crypto protocol design is *hard*

How Does WPA2 Fare in our Goal Model?

Host Access: Blocked

Network Access: Blocked

Content Access: Blocked

Metadata Access: The source and destination MAC addresses are sent in the clear

But…

Metadata Access

Who would want to exploit metadata? Remember that it's MAC addresses, which stay local.

- Intelligence agencies *love* metadata, to see who talks to whom
  - They also may have or be able to build a database of MAC addresses
- Network operators can track it
  - Intrusion detection; marketing (for public WiFi nets)
- For pay networks, impersonate someone else's MAC address and run up their tab

WPA2 WiFi versus Wired Nets

- With wired nets, you have a well-defined perimeter
- You also have switch ports to localize misbehavior
  - Suppose that an internal machine has been hacked and starts spoofing its MAC address and IP address
  - On wired nets (with enterprise-grade managed switches), you can see which physical port the spoofing is coming from
  - With WiFi, the attacker could be more than a kilometer away
  - Also: switches (mostly) direct traffic to the intended machine; with WiFi, everyone on the same access point will see it
  - Similarly, ARP-spoofing without detection is easier
- But: the encryption with WPA2 Enterprise is per-user, so other on-net nodes can’t read the WiFi traffic; they can on some wired nets

Tracking Misbehavior on WiFi

- Start from the access point
- Using radio direction-finding is harder than you would think—problems with multipath
- Block-list the offending IP and/or MAC addresses and see who complains
- That won't do much good against a serious attacker!

Should We Worry About This?

- “Flaw in billions of Wi-Fi devices left communications open to eavesdropping” (https://arstechnica.com/information-technology/2020/02/flaw-in-billions-of-wi-fi-devices-left-communications-open-to-eavesdroppng/)
- When a device disassociates from an access point, remaining traffic is sent encrypted with a key of all zeroes—and it’s possible for the attacker to force disconnects
- Is this scary?

Not Really

- It’s at best access to a bit of content and a bit of metadata
- The attacker can’t control what’s made available

What About Public WiFi?

- Encryption is almost never used
- If it is used, it’s WPA2 Personal, not Enterprise, so there’s no protection against on-net eavesdroppers
    - Remember that most public WiFi nets are used by normal people, not computer geeks
    - Asking users to put up with crazy configuration options *will not work*
- What are the risks? The attacker can achieve all of our goals. Is this a serious problem?

Public WiFi: Content Access

- Content is obviously available; use of encryption is *mandatory*
- Better yet, use a VPN, to encrypt all traffic leaving your computer
- MAC address metadata is always sent in the clear—but VPNs hide your destination IP addresses
  - If your MAC address is sensitive, change it-–you generally can—before connecting to the WiFi network
- Is your VPN gateway sensitive? If so, use Tor
  - VPN gateway addresses are most likely to be of interest to intelligence agencies

Public WiFi: Network Access

- It’s a public net; no barriers to joining…
- Some nets, e.g., in hotels, *may* restrict access, but via a very low barrier

Public WiFi: Host Access

- Some public WiFi nets prevent hosts from contacting other hosts on the same network, but you can’t count on that
- This is the hard question: what is the risk *to your computer* if you use a public WiFi network?
- How do we analyze this? Of course: execution environment

Are We Missing Something?

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Link
1	Physical

The Network Stack

7	Application	Email, HTTP, etc
6	Presentation	Presentation
5	Session	Session
4	Transport	TCP, UDP
3	Network	IP
2	Link	WiFi, Ethernet
1	Physical	Radio, cable, fiber

Link Layer Issues?

- We usually think of security problems from layer 3 up
  - But eavesdropping is often a layer 1 issue
- What can happen at layer 2?
- For WiFi, there's a protocol for associating with the network and for negotiating a cryptographic key
- Are there problems there? Maybe!
- (N.B. Apple does a lot at layer 2, e.g., Airdrop)

Execution Environments

Corporate Desktop

- Mostly friendly hosts
  - But what if some internal host has already been hacked?
- Good internal monitoring should detect traffic diversion attempts
- A number of services on, to permit collaboration
- Attacker goals
  - Access rights of this computer
  - Data stored on it (but maybe not much)

Public WiFi

- Mostly unknown hosts
  - Some might be evil
- Intercepting traffic is easy
- Few externally-facing services are needed—unless you need to talk to a public printer
- Attacker goals
  - Access rights of this computer, *especially after it goes back to the office*
  - Data stored on it (probably more than at work)

Differences

- There are more assets at risk for laptops on public WiFi
  - It may be how there is an inside machine that was hacked—it had been an exposed laptop…
- For a serious enemy, the odds of an already-hacked inside machine are moderately high
  - *No incremental risk of attack!*
- If we can turn off some services when outside—better yet, have them turned off automatically—the risk of attack may be *lower* outside
- If the corporate firewall works well—an assumption!—it can block nasty stuff from "recreational" sites employees might visit

Conclusions

- The incremental risk to laptops is not high, especially with modern operating systems
- Disabling some services automatically is a good idea
- Use a VPN
- Encourage use of the corporate firewall, even for recreational browsing

Fake Hotspots

- Most devices *automatically* associate with known nets
  - Nets are identified by SSID (Service Set Identifier)
- What if an attacker spoofs a known net, corporate or hotspot?
- Conclusion: *always* use bilateral authentication
  - Software should *always* check the validity of the far side's certificate

Always-On VPNs?

- Can we have an always-on VPN?
  - Recall that a proper VPN will reject non-VPN packets
- The problem is sign-on—many public hotspots require some sort of sign-on page before they let you out to the Internet
- How do you protect the browser that does the sign-on?
- Sandbox it?

A Sandboxed Browser?

- Browsers are sandboxed; should they be allowed to bypass the VPN?
- But browsers are always sandboxed because of how vulnerable they are
  - (Is the sandbox secure?)
- And: browsers often have access to stored passwords
- What's needed: a separate browser that's outside the VPN, with only the passwords needed to connect to networks
- **PLUS:** a VPN that is automatically and always started for all other network connections
- That *should* make public WiFi safe

Doing Without WiFi?

- Should companies just do without public WiFi?
- Why do employees travel with laptops? *To increase their productivity—they* need *connectivity*
- In other words, there is a risk from no WiFi as well

Cellular Wireless

- Cellular service is usually safer
- It’s relatively easy, even for high-end hackers, to divert calls to a mobile phone
- But data is data, and is sent over IP, which isn’t controlled by SS7
- There are still all of the usual IP routing games, but that’s an Internet story, not a cellular one
- However…

IMSI Catchers

- Fake base stations (sometimes called “Stingrays”, after one popular model)
- Can locate cell phones belonging to targets; can also intercept traffic from them
  - But: must be close enough to the target to present a stronger signal than the real base stations
- Newer mobile phone protocols authenticate the base station, too
  - But: what if the enemy controls the real cellular network?
  - In many countries, there are PTT—postal, telegraph, and telephone—ministries, i.e., the phone network is operated by the government
  - Or: phone switches can be hacked

IMSI Catchers: Uses

- Primary law enforcement uses:
  - Is a given number in a given location?
  - What numbers are in that location?
  - N.B. Like all base stations, IMSI catchers have a finite radius they can reach, and almost certainly less than a real base station due to lack of a good, high-mounted antenna
- Can IMSI catchers wiretap calls? Data? They could for 2G cellular, but that's *old*. Can they today? Unknown publicly.
  - But: on their home turf, a foreign intelligence agency can play its games on the land side; they don't need IMSI catchers for that
  - What about intelligence agencies operating in other countries? There have been claims about many IMSI catchers around D.C., but no proof

Attacker Goals: Cellular

Network Access: Trivial

Host Access: Generally blocked by carriers, but not always

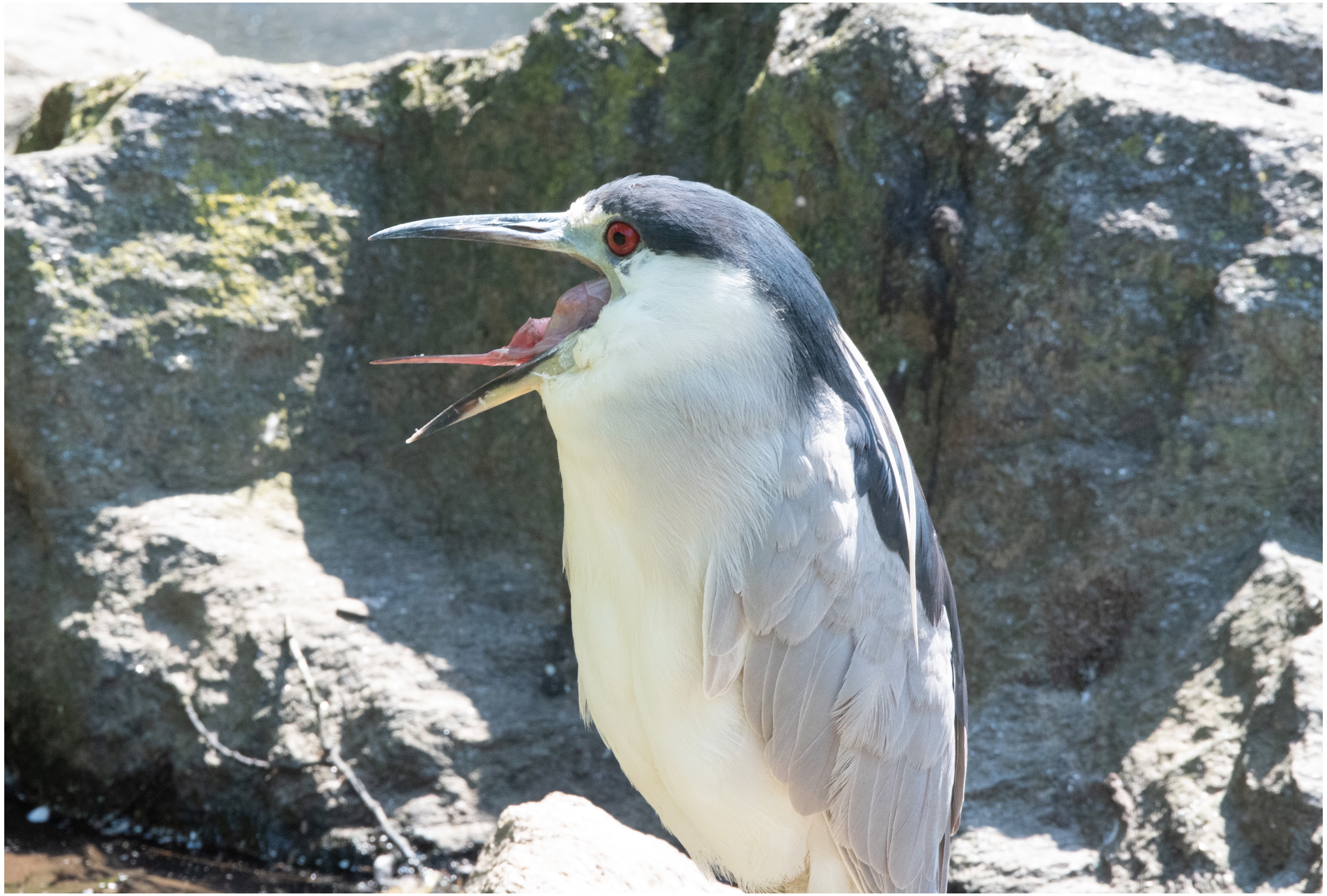
Content Access: Encrypted over the air; how strongly is not clear

Metadata Access: Some available via IMSI catchers

Is Wireless Safe?

- “It depends”
- What is your threat model? Who are your enemies, and what are their goals?
- Non-cellular nets require proximity, which limits attackers
- Cellular networks are safer *except* when dealing with intelligence agencies
- And again: what is the cost of being offline?

Questions?



Black-crowned night heron, Central Park, July 23, 2021