Security 12: Wireless and Wired Security

i.g.batten@bham.ac.uk

Wired Networks

- You know where the wires are (you hope)
- Tapping wires requires significant effort and is detectable
  - “Tempest” attacks are extremely difficult, although not impossible, as UTP and (more so) STP cable doesn’t radiate very much
  - Fibre available to reduce Tempest risk to essentially zero.
- Paranoid sites can use time-domain reflectometry (TDR) to find cuts and splices in cables and fibres, and optical power monitoring to find bends.
  - Cheap testers won’t find splices, but you pay more, you get more. You can find open circuit faults with a £600 TDR tester. Renting a Megger Teleflex SX to map splices is £600 a week.

Wireless Networks

- None of the above
- Unlikely to be limited to your site, and certainly isn't limited to your site if attacker has high-gain antenna.
- In 2017 a wire only carries one client's worth of the site traffic, while a wireless network carries much more, so benefits of interception potentially greater

And insiders gain too

- Switched wired networks make insider (people with legitimate access to the cable plant) attacks harder: they need to tap either more cables, or cables closer to the core of the network and therefore ideally more secure
- A wireless network offers traffic between multiple pairs of systems without leaving your desk: have receiver, have keys.

Multiple problems

- Access control: don’t want attackers on the network able to send and receive traffic as end-stations themselves
- Integrity: want packets to be unchanged in flight and to come from where they purport to come from
- Confidentiality: don’t want attackers to be able to “sniff” traffic and read its contents
    - Note: not always necessary to be able to read contents to make changes useful to an attack

Reality Check

- Wired networks have some physical security, but it isn’t great.
- Most wired networks aren’t proof against someone putting their own kit on the end of a wire.
- Most networks don’t protect against physical attacks on the plant
  - Ideally routine TDR, cables in pressurised tubes, cables on walls, fibre for backhaul, etc.
  - Reality is dense mess under the floor

So questions are:

- Is our wired network secure enough?
- Is our wireless network as secure as our wired network, or secure enough?
- “enough” is the product of a rigorous risk assessment or, perhaps, a damp finger in the air.

Wired Network Security

- MAC filtering
  - Switch only accepts traffic from a specific set of MAC addresses on a specific port, or…
  - Switch / Network only recognises devices with specific MAC addresses

MAC Filtering: Little Value

- Will stop some configuration errors, and might prevent users moving equipment around and making a mess of your patching records
- Low-skill attacker can change MAC address of attack system to match authorised system
- Even if not supported by ethernet card (in practice always is) this is standard part of virtualisation stack for bridging.

Why do people MAC filter?

- Habit, or ritual
- Does make it easier to sack people, as they have to make an overt move in order to perform their attack (like having “No Trespassing” sign on fence). **This can be very useful.**
- Reveals assumption that controlling equipment controls capabilities, which isn’t really true

802.1x Authentication

- Client device supplies username and password or proves ownership of certificate (via various mechanisms) which unlocks port it is connected to
- Roughly same protocols as used to do wireless authentication, in fact
- Pre-authentication, port is either restricted to authentication only, or is connected to restricted VLAN.
- “Posture Analysis” allows running of checks on patching and virus scanner levels, with re-mediation

Problems with 802.1x

- User interfaces for supplying credentials early in boot can be hostile
- Difficult to generalise to printers, servers, Web Cams…
- Are you authenticating the user (they can supply their username/password pair to any device, including their own) or the device (where do you keep the keys so hostile users can't borrow them and put them on their own phone)?
- In early implementations defeated by a switch at the user end of the cable (device 1 authenticates, port opens, device 2 can also access the network)
- More recent implementations also MAC Filter, but can be attacked with carefully timed swapping of MAC address while attack switch holds link up
- Look up "multiple supplicant support" to see how complex it is to deal with multiple devices all trying to authenticate via a simple switch
- TPMs help key management, but not much (sadly, a universal truth)

Benefits of 802.1x

- Again, makes it easier to sack people by forcing misusers to perform “overt acts”.
- “Posture Analysis” can enforce patching in co-operative environments, although it is not obvious that the complexity is justified.
- Probably better solved with good host patching regime, but possible that 802.1x (fashionable 2005–10) plus BYOD (fashionable 2011–date) overlap.

Problems with 802.1x

- Trivially bypassed in many cases, and hard to assure an installation
- No encryption of payloads, so an attacker who can patch into cables is no worse off
- Key management for non-personal devices is tricky
- Key management for personal devices is tricky
- Personal view: very few implementations successful and on-going

Wireless Networks

- First attempt, WEP: “Wired Equivalent Privacy”
- Intended to make a wireless network about as secure as a naive wired network

WEP

- Everyone has a shared key (40, later 104, bits)
- A 24 bit initialisation vector is generated, so total 64 or 128 bits of key material
- It’s used to encrypt packets, using the RC4 algorithm
- All stations with the key can see all packet contents: behaves like old-fashioned ethernet

WEP is Broken

- WEP is trivially broken, and is an object lesson in why you should not attempt to roll your own crypto.
- Attacker can attempt to obtain key, or can attempt to obtain evolved key stream
- Additionally, generating IVs is very difficult in embedded devices

RC4

- RC4 uses a key and an initialisation vector to generate a evolved keystream of bytes, which are XOR’d with the message to form cipher text.
- Means that you have two possible attacks: recovering the key, or recovering the evolved keystream

Keystream

- Trivial to inject known packets into the network, watch them emerge encrypted by access point, and obtain portion of evolved keystream
- Generating unique initialisation vectors extremely difficult at power-on of embedded device (no hardware RNG, air turbulence in disk drives main source of randomness in computers, packet arrival times known to and potentially controlled by attacker)
- In many cases, IVs were drawn from very small space, so number of distinct keystreams was very small

Details of Attacks

- Fluhrer, Mantin and Shamir recovers keys
  - Amusingly, RC4 = Ron's Cipher 4, Ron = Ron Rivest. Shamir = Adi Shamir. So that's the R and the S from RSA public encryption, here on opposite sides of the RC4-breaking game
- Plus a variety of other "practical" attacks on weak use of RC4 (malleability attacks, in particular)
- WEP: **Completely Broken**

WPA

- Emergency interim fix to WEP, adding sequence numbers, better key mixing, integrity checks
- Idea was that it could run on same hardware as WEP, so only needed a firmware fix: doesn’t involve box swapping
- Mostly of historical interest
- Still weak

WPA2

- Proper encryption for grown-ups: third time lucky
- Initial secret (either PSK or result of per-user login) used to generate a transient key used just for one session (~1 hour)
- Underlying encryption is AES128
- Used correctly, can be accredited to IL4 CONFIDENTIAL in old money, possibly even low end of SECRET under new system (higher would require AES192).
- Needs new hardware
- You can’t see other people’s traffic (special extra key for broad/multicast traffic).
- Key management complex, risky and difficult to do correctly: using WPA2 for more than access control probably not for the casual user.

WPA2 Authentication

- If you have a crypto background, look it up
- Basically, each party generates a random, sends it to other party, combines with shared key and generate another key. If the keys agree (ie, encrypt and decrypt identically) then both sides have the shared key and can communicate.

WPA2 PSK

- Users share a common pre-shared key
- Key derivation function (PBKDF2) repeatedly iterates a strong hash function to generate better key material, maybe.
  - Mixes in SSID, so benefit to having an SSID no-one else is using is forces dictionary attacks to start afresh for you (random works well)
- Compromise of key exposes all past recorded traffic and all future traffic: no forward secrecy on session keys
- Compromise of key requires global change in order to lock out attacker who has old key
- Only usable for small, trusted groups of people (although in practice businesses use it more widely)
- Someone who captures a key exchange and has the pre-shared key can read traffic on that connection (special case of lack of forward secrecy)
  - Doesn’t use Diffie-Hellman
- KRACK (allows resetting of connection, now mostly patched)

WPA2 Enterprise

- Users authenticate individually to authentication server (usually using “Radius”) and a per-user key is negotiated (with some extra magic for broad- and multi-cast).
- Can interwork with SSO/AD/etc type systems
- Derived keys are per-user, so compromise of user key only compromises their traffic
- eduroam shows this working world wide, but…
- …mostly for authentication, not confidentiality

WPA2 Enterprise

- It’s also worth noting that a lot of equipment doesn’t support it, because it’s tricky to implement and requires quite a lot of GUI support, testing and so on.
  - WiFi radios, televisions, set top boxes, gaming systems, etc, etc, etc.
  - This limits applicability in hotels, halls of residence, business parks / shared workspaces.
- There’s also the issue that enrolment is awkward, as users need to be enrolled in whatever backend authentication system is being used.
  - This is true of so many things that “enrolment is awkward” should be a tee-shirt or something.

Wireless Security Objectives

- Authentication
  - I don’t want the wrong people using my network, I want to be able to identify people who are using my network, but if they need confidentiality or integrity that is their problem.
  - Hotel/Coffee/etc network
  - For practical purposes, “any network you do not personally control”.

Confidentiality

- I want people using unencrypted applications to be able to work without being spied on by a wireless eavesdropper
- More corporate, where there are a lot of old applications which can’t easily be SSL-ised
- I may not need forward secrecy, because I am not worried about people recording encrypted traffic to decrypt later if a key leaks (perhaps this is not a good assumption).

Integrity

- I don’t care if users’ traffic can be read, but I don’t want their traffic to be routed elsewhere (ie, make MITM attacks as hard as possible)
- Opens up issue of Secure DNS: attacker can provide fake addresses for known names, and then play games with redirects.

WPA2

- Strong for authentication: no known attacks better than brute force (but see next slide)
- Worrying for confidentiality, because of lack of forward secrecy. Certainly, WPA+PSK is not sufficient for anything more than a family (small number of users, nothing wildly secret). WPA Enterprise better, but still not great, and complex.

Warning: WPS

- Wireless Protected Setup
- “Press a button, get a key”, but also offers authentication using PINs
- Frighteningly insecure: should not be used, ideally should be disabled (although some systems disable it without actually disabling it)

WPS Error

- If a PIN has eight digits, requires $10^8/2$ guesses on average
- If system will verify PIN four digits at a time, requires $10^4/2 + 10^4/2 = 10^4$ guesses.
- Reduces security from n to sqrt(n) operations
- And in fact, in the 8 digit PIN the last digit is a checksum, so $10^4/2 + 10^3/2 = 5500$ guesses on average.
- Again, don’t roll your own crypto…

Again, in passing

- Generating a random number on an embedded device is difficult without hardware assist, especially close to power-on before serious traffic has passed.
- An attacker who can observe all traffic can probably massively reduce search space: where else is the entropy coming from?
- Pro tip: to sound informed in a security meeting, “where is the entropy coming from?” is always a good question.
    - Another tee-shirt.

So, wireless security

- Option 1: use the security your wireless systems have
- Option 2: assume the wireless is open, use a VPN
- Option 3: option 1 for access control, option 2 for confidentiality and integrity

VPN

- Details in a later lecture, but a VPN (“Virtual Private Nework”) offers confidential communications over an untrusted carrier, using arbitrary authentication and encryption between end points.
  - Can easily offer full forward secrecy, two factor, etc
  - Extensively analysed products, mostly built on well-understood underpinnings (TLS, IPsec).
  - Available in trusted/accredited/etc versions up to SECRET and beyond (although these solutions are standalone hardware, as the keys can’t be allowed into end-user computers).
- VPNs offer high confidentiality and high integrity, provided you are not too bothered about availability: you can always jam wireless given enough resources, and frequency-agile and/or spread-spectrum jam-resistant networking requires exotic, exotic licensing you can’t get unless you are military.

Rogue APs

- Common meme: “don’t do banking in hotels”. “Don’t do banking in Starbucks”.
- Evidence for efficacy of these attacks at scale is not great: they rely on poor user behaviour and are quite complex to execute.
- But the place to defend against this is in browsers and in websites; attempting to make all public WiFi secure is impossible.
- And there are other benefits: makes compromise of home/workplace WiFi less serious if assumption is that it is broken anyway.

Rogue AP

- Option 1: assume users access bank via Google search for name. Catch lookups of Google A records or TCP connections to known Google addresses, hand back results with fake URLs for bank (110ydsbank, that sort of thing).
- Protections include https, certificate pinning, user education, EV certificates, etc.
- Harder (but not always much harder) to pull off since Google went https, depending on browser behaviour.
- I have seen some hotel/shop networks which try to force you to use unencrypted Google, presumably as part of doomed content-filtering mechanism.

Rogue AP

- Option 2: Fake DNS records for mybank.co.uk
  - use it to serve fake certificate and hope user does not notice; or
  - use it to push user to site which redirects to mybank.co (or something else visually similar) for which you have a real certificate; or
  - use it to push to site which redirects back to unencrypted version and hope user does not notice.
  - Will be caught by certificate pinning, HSTS and so on.

Remember

- We often dismiss potential attacks on the grounds that skilled opponents would not be caught by them.
- However, attacker does not need to get lucky every time, or even most times, they just need to get lucky occasionally.
- As people who notice and avoid the attack don’t have an obvious way to report it, or share it with less skilled users, the attacker can keep on trying: the failed attempts cost them nothing.
- If you went to the barista in a Costa and reported that there was something awry with the wireless network, what do you think would happen? My guess is “nothing”. Would you have the energy to take it further?
- All that said…

Rogue AP

- Much discussed, not much evidence of wide deployment against passers by.
- Probably used for targeted and focused attackers who know what they are doing and don't get caught.
- Absence of cryptographic protection on initial DNS lookups very scary, probably more exploits to come
- Good argument for VPN: if you're using a wireless network you don't trust, connect to a VPN which you do trust (I use privatetunnel.net, as well as my home server).

Public Service Announcement (1)

- HSTS: HTTP Strict Transport Security, **RFC 6797**
- HTTPS header which says “for the next XXX seconds *(31536000, one year, is good)*, this website will always be HTTPS, and you should reject any attempt to serve it **un**encrypted. Cache this fact”. Means attacker must catch **first** access to site by browser installation/user pair.
- HSTS Preload into Chrome and derivatives mean attacker can never force use of unencrypted version.
- **Please, please, please convince your managers to use this.**
- `add_header Strict-Transport-Security "max-age=31536000";`

Public Service Announcement (2)

- HPKP: HTTP Public Key Pinning **RFC 7469**
- HTTP header which says “this site will always use one of these public keys, until at least this time; cache this fact”. Stops fake certificates stone dead.
- Tricky to set up as requires you advertise future public keys in case of revocation/compromise, hence need to keep offline copies of private keys. Could be tricky if hardware storage modules are involved. Possible DoS issues (recovery from compromise is hard).
- In 2018, looks dead: not going to be implemented in Edge or Safari, isn’t implemented in Firefox, is slated for removal in Chrome over next few months.

So…

- Use wireless authentication to keep out people you don’t want using the infrastructure, but don’t trust it for confidentiality (no forward secrecy, probably no one-time passwords) and encourage users to take their own precautions based on their personal risk appetite.
- Layer a VPN over the top to provide desired level of confidentiality
- Proper VPN boxes are much cheaper than they used to be.

And of course…

- If you're able to justify running a full-on VPN on the wireless network, you have to ask whether you should do something on the wired side as well.
- IPsec is a later lecture

Summary

- Wired networks have some security, but you can add more; 802.1x isn’t great, but there’s IPsec or VPN technologies if you really need it.
- Wireless networks are wide open, or at least best treated as such, and if you’re doing anything of even passing confidentiality you need either IPSec/VPN or to be confident in the encryption of your traffic.
- IPsec and other VPNs are well analysed and available in assured/accredited forms.