What I Wish People Knew About Networking

Ethernet and IP Networking From the Ground Up

Adam Thompson – athompso@athompso.net

Font sizes

are

all over the map

deal with it

(sorry)

Thanks, PowerPoint!

What this talk covers

- How networking actually works
- Where/when/how it doesn’t work
- Where does Hanlon’s Razor apply?

Hanlon's Razor

- "Never attribute to malice that which is adequately explained by stupidity" – Robert J. Hanlon, *Murphy's Law Book Two* (disputed)
- "Cock-ups, not conspiracies" – attribution unclear, common British usage
- "There is very little deliberate wickedness in the world. The stupidity of our selfishness gives much the same results indeed, but in the ethical laboratory it shows a different nature." – H.G. Wells, *The Wheels of Chance*

Hanlon's Razor, con't

- Stop believing that EVERYTHING is an attack.
- ALMOST everything that COULD be an attack, is actually an accident, a mistake, or a screw-up, of some sort.
  - An accident *can* be an attack, e.g. an accidental DoS/DDoS, although they're *usually* self-inflicted.
- Assuming there's an attacker hiding behind every door when 80% of the doors *can't even exist* leads to inefficient, needlessly-expensive mitigations and even *prevents* proper network and systems design.

It's ***Just*** Math and Boolean Logic

- OK, that sounds scary to some people, but it's really not.
- Binary math
  - (oops, now it's more scary, not less)
  - …that I can explain in one slide. So there.
- Binary logic
  - Two slides. It's not hard.

- If you can add, subtract, multiply and divide, you can do this.
- If you can round up to the nearest dollar, you can do this.

Binary Math

- Everything in networking is based on binary math.
- "Mastering" binary math is just as hard as "mastering" regular decimal math. We don't need to "master" math in *any* base, we only need to understand a couple of fundamental things.
  - We usually only need to worry about positive whole numbers between 0 and 255; or between 00000000 and 11111111. This drastically simplifies the problem space.
- Powers: $x^y=z$ means multiply x by itself y ("exponent") times. (You'll also see $x\text{^}y$ or $x**y$, since ASCII doesn't do superscript.) Two special cases exist:
  - If $x=0$, then $z=0$.
    - Otherwise, if $y=0$, then $z=1$.
- Numbers in any base are written in a series of powers, where the exponent starts at 0 on the right-hand side and goes up by 1 as you move left:
  - Base 10 (decimal):
    - $21 = 1 \times 1^0 (=1) + 2 \times 1^1 (=10)$
  - Base 2 (binary):
    - $10101 = 1 \times 2^0 (=1) + 0 \times 2^1 (=2) + 1 \times 2^2 (=2*2=4) + 0 \times 2^3 (=2*2*2=8) + 1 \times 2^4 (=2*2*2*2=16)$

“All on one slide” (I didn’t lie, exactly…)

I kind of cheated to fit it all on one slide. Let’s revisit that core concept, in larger print:

- Base 10 (decimal) example:
  - $7421 = 1 \times 1^0 (=1) + 2 \times 1^1 (=10 =10) + 4 \times 1^2 (=100 =10\times10) + 7 \times 1^3 (=1000 =10\times10\times10)$
- Base 2 (binary) example: [it’s 21, in decimal]
  - $10101 = 1 \times 2^0 (=1) + 0 \times 2^1 (=2 =2) + 1 \times 2^2 (=4 =2\times2) + 0 \times 2^3 (=8 =2\times2\times2) + 1 \times 2^4 (=16 =2\times2\times2\times2)$

Binary Logic

- Binary logic doesn't have a good equivalent in decimal
- It does correspond to the *physical* world, however.
  - Electric & electronic circuits are based on binary logic
  - ***Plumbing*** and piping and valves exhibit binary logic!
- This is one of the the "logics" that comes out of Philosophy departments. Boole wasn't designing circuits!
- If-this-and-that-then-something
- If-that-and-other-then-this
- Etc.

Binary Operators / Operations

- Only 4 operations:

BINARY "NOT"

```
NOT 0 = 1

NOT 1 = 0
```

BINARY "XOR"

Irrelevant

BINARY "AND"

```
0 AND 0 = 0

0 AND 1 = 0

1 AND 0 = 0

1 AND 1 = 1
```

BINARY "OR"

```
0 OR 0 = 0

0 OR 1 = 1

1 OR 0 = 1

1 OR 1 = 1
```

Multi-digit Binary Logic

- When you perform a logical operation (AND, OR, NOT, etc.) on a binary number, just apply it to each pair of digits independently:

Multi-digit Binary Logic


```
ab OP cd = (a OP c),(b OP d)

e.g. NOT 11 = (NOT 1),(NOT 1) = 00

e.g. 00 AND 10 = (0 AND 1),(0 AND 0) = 00

e.g. 010 OR 011 = (0 OR 0),(1 OR 1),(0 OR 1) = 011
```

Failures / Attacks

- Failures:
  - 99.999% Human error
    - How do you attack math?
    - But, LOTS and LOTS of human error, oh boy!
    - Incorrectly-designed or incorrectly-implemented circuits
  - Failing circuits
    - Electron migration; Static (EMP); Physical damage; etc.
- Attacks:
  - How do you attack math?
  - Exploiting known errors in circuit implementations
    - Pentium DIV bug?
    - MMU bugs could have real-world attacks
    - How does the attacker get to the electrical signals on the chip, though? Not impossible, but insanely hard.
- Mostly out of scope – if there are problems at this level, we can't do anything about them except buy different hardware.

Organization: Layering & Encapsulation

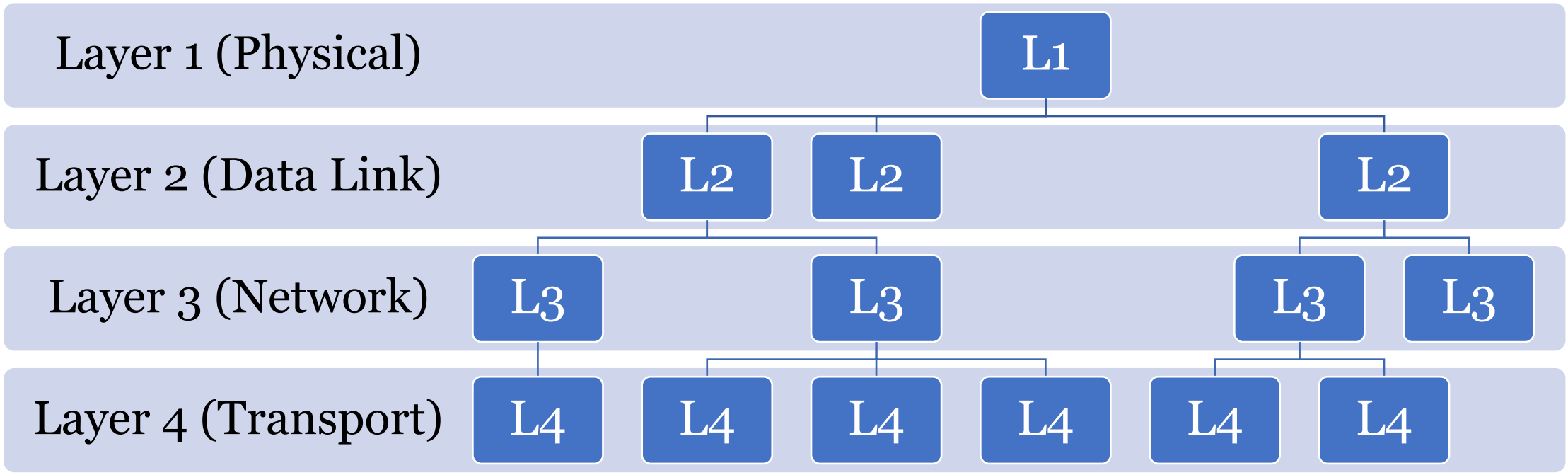
- Networking is ***very*** strongly organized into layers
- You can also – mostly – think of the layers as encapsulation instead, if that helps; both approaches are mostly correct.
- "Layering violation" is a phrase used to indicate a particular system is behaving in an unacceptable way, even though it *works*. Think "redneck engineering": it works, but it ain't *right!*
- The OSI stack is hugely problematic, mostly obsolete, yet is still the gold standard reference. We'll only be talking about some of its layers.

OSI Stack

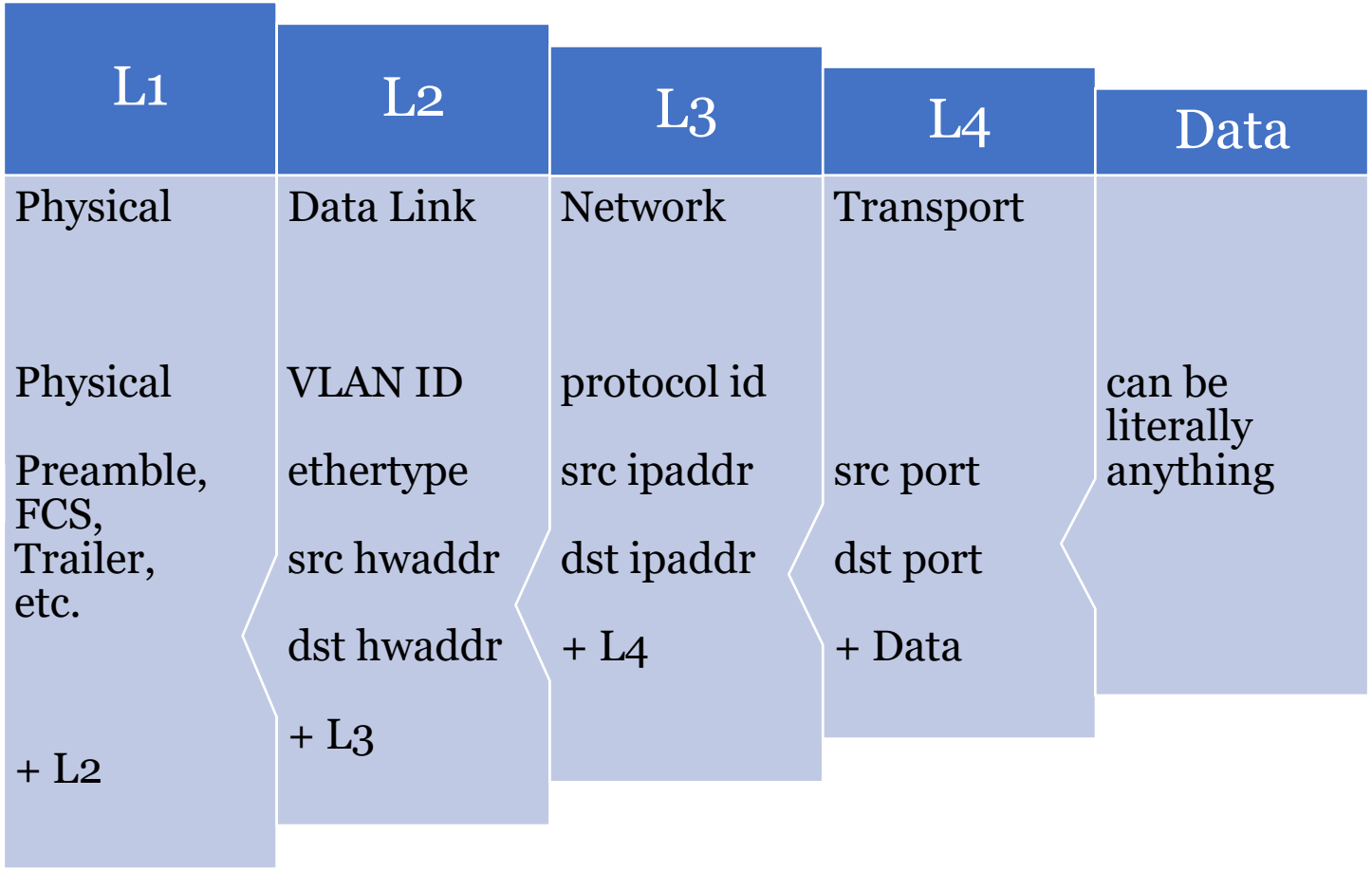
We really only care about the **bottom 4 layers**, and of those only a few things are still common today.

1. Physical Layer
    - Fiber optic cables
    - Copper cables (Cat5, etc.)
    - Radios, including both point-to-point and WiFi
2. Data Link Layer
    - Ethernet
    - Radio/wireless protocols that emulate Ethernet
3. Network Layer
    - IP (and its related protocols)
4. Transport
    - TCP ports
    - UDP ports

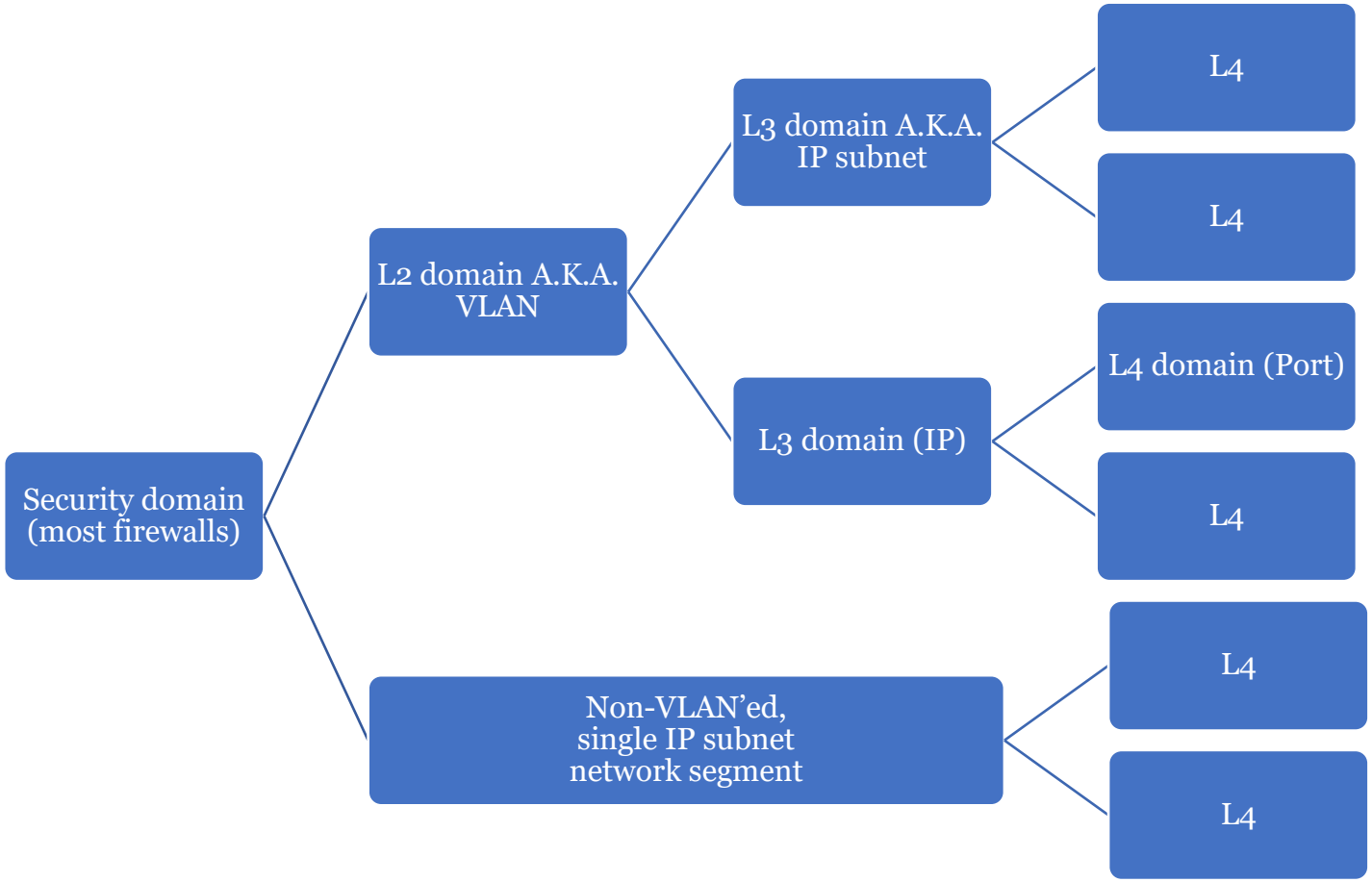
OSI stack, as hierarchy



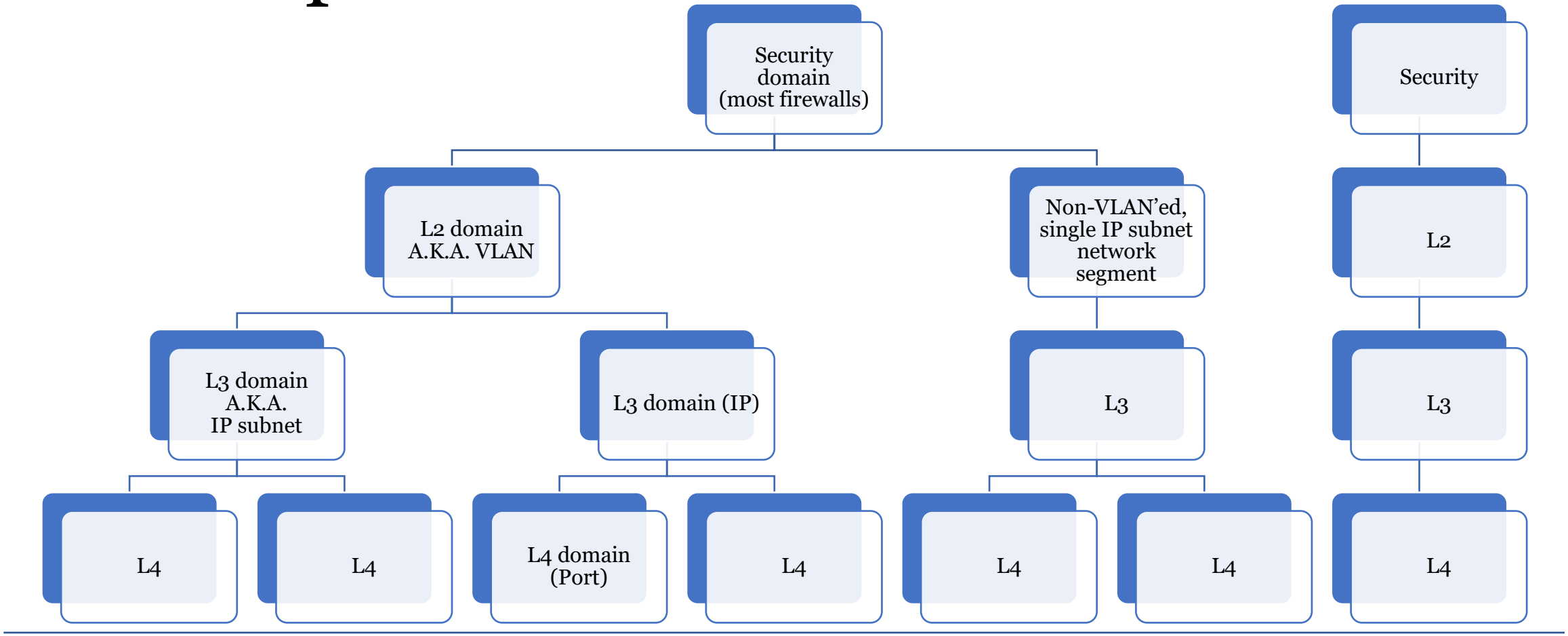
OSI stack, as encapsulations



Layers vis-à-vis Security



The security relationship is critical. Let's repeat it:



Layering attacks

- Layering attacks do not exist. Layers are a concept, not a thing.
- Layering violations exist all over, everywhere you look. These are sometimes deliberate, but not malicious.
  - Unless you count the vendor taking your money and doing something that stupid, on (or to) your network, as “malicious”. YMMV…
- Many network-centric attacks focus on layer boundaries because the lower you go, the more unhandled corner cases you can find, and the fewer devs who understand that layer there are to fix bugs.
  - Much of the internet only works “by convention”, despite the best efforts of the IEEE, IETF, et al.
  - “Just because you can, doesn’t mean you should”
  - “Just because you can, doesn’t mean it’ll work”

Hanlon's Razor

- Layering violations
  - Happening always,
  - Happening everywhere,
  - Because incompetence abounds!
- Layer attacks
  - Aren't really a thing. How do you attack a concept?
- Attacks focused on getting to a lower layer?
  - Lots of soft spots with physical access
    - Extremely difficult to detect
    - Basic physical security remains the best defense
  - Remote (software) attacks exist, but are well into APT territory
- If you can't tell if it's an attack, this is perhaps the only place I would agree with defaulting to "suspicious" instead of "oh, those dumbasses".

Layer 1 - Physical

- Copper
- Fiber
- Wi-Fi
- Barbed wire fences
  - (this really happened, at least twice)
- etc.

Physical Layer - Copper

Never assume your cable or patch cord “must” be OK, even if it’s brand new out of the packaging or freshly installed!

Gigabit Ethernet – Wikipedia explains the nuts and bolts better than I can!

Typical copper concerns:

Cause	Effect	Detection	Temporary Fix	Permanent Fix
Stretching, folding, bad crimping, crushing, corrosion	CRC errors, silent retries & retransmissions	Interface statistics (via WMI or SNMP)	Replace the cable or re-terminate.	Only use patch cords with factory-molded boots. Use only BICSI- or AMP-certified cables and installers.

Copper Faults & Attacks

- Statistically 100% of every copper problem, ever, was a fault, not an attack.
  - We'd need a *lot* of significant digits to measure the attacks
- However, it's easy to tap copper with the right equipment.
  - Again, physical security is the solution.
  - You don't let people near your ethernet jacks unless you trust them, right? Right?? RIGHT???
- Disabling the switchport does not mitigate attacks on wiring, and prevents detection of the attack.
- Physical security is hard and expensive (and a giant PITA), so most people just don't do it. This is usually your soft underbelly.

Hanlon’s Razor

- It’s *always* a fault in the copper.
  - Seriously, even if you’re running a compromised CIA post in Moscow, it’s always just a fault in the copper.
  - …except for that one time it isn’t. Good luck figuring this one out.

Physical Layer - Fiber

Never assume your cable or patch cord "must" be OK, even if it's brand new out of the packaging or freshly installed!

10 Gigabit Ethernet - Wikipedia explains the nuts and bolts better than I can!

Typical fiber concerns:

Cause	**Effect**	**Detection**	**Temporary Fix**	**Permanent Fix**
Flipped fibers	No light (AKA -40 RX dBm)	No port LED, interface remains down	Flip the fibers, *in only one place*.	None. Pre-flipped patch cords exist, YMMV.
Dirty ends	Insufficient light (RX dBm values too low)	Low signal strength, retransmits, CRC errors, etc.	Clean both SFPs, all 2 or 4 patch cord ends, and all patch panel ports.	Use protective dust caps. Be careful.

Fiber Faults & Attacks

- Faults are easy to detect
- Attacks are nearly impossible to detect
  - OTDR is your only (affordable) friend
  - Good news if you're red-team, I guess?
- Physical security helps, as usual
- Encrypted bitstreams help protect the confidentiality of your data, but are overkill for almost everyone everywhere.
  - L2 encryption in 2024 is called "MACsec", and is *reassuringly expensive* as well as often reducing the hardware's overall capacity

Hanlon's Razor

- If you can detect the fault, it's probably not an attack.
  - Yeah, I know, that was helpful, wasn't it?

- OTDR testing is highly invasive, taking the line out of service to test, but can reveal any unexpected interface in the fiber.
  - Most of those are just a rush job by the installers, the splicers, the carriers, etc.
    - Sometimes old age causes "bumps in the road" too, no matter how good it was when installed, 30+ years ago.
  - When you see something unexpected on the track, it's probably just a patch panel or splice box you didn't know existed.
  - But if you can't identify it… get that checked out!

Physical Layer - Radio

- Even radios can be dead or flaky “out of the box”.
- Antenna lobe patterns – know what they are, and what your radio’s patterns are. *Tilting* an AP to improve (or reduce) coverage is a thing!
- Interference – glass is (surprisingly?) worse than metal, which is worse than brick, etc.
- Roaming – turn it on at the AP; it will help you
  - If it hurts you instead, you’ve done something *else* wrong
- Controllers – use them to allow roaming to work better (and to save your sanity when making changes to multiple APs at once)
- Turn your radio power *down* and watch performance *increase* in dense environments!

Radio Attacks

- Radio is a **broadcast** medium. A *shared, public* medium.
  - Even when using point-to-point radio, the Fresnel zone is usually fairly big
- Using radio for L1 is possibly the most insecure way to send information other than smoke signals or taking out billboards on highways.
- Every sane radio standard includes encryption somewhere below L2. Many (e.g. 802.11) have their own entire protocol stack that lives within the OSI's Layer 1, from the perspective of TCP/IP!
- Snooping is trivial
- Denial attacks are trivial

Hanlon's Razor

- Most radio problems are just bad radio network design
- Denial-of-service attacks are, however, frequent, because you need almost zero skill, and nearly zero resources, to perform them
- DoS attacks and the spectrum management features built into some higher-end Access Points are nearly indistinguishable
- There are many attacks against the sub-protocols *inside* radio protocols
- Unless you're specifically looking for attacks, you likely won't find any (except DoS)
  - Even those attacks are likely self-inflicted, or "innocently" inflicted by neighbouring radio systems

L2 – Data Link

- Ethernet
  - and…
  - um…
  - Everything else just emulates Ethernet now.

- Some other legacy tech still exists, but was always niche, now vanishingly rare.

Data Link Layer - Ethernet

- Every Ethernet (or Ethernet-like) packet has some stuff, some headers, some data, and some more stuff. See Ethernet frame – Wikipedia for more details.
- We don’t care about the leading and trailing stuff.
- We don’t even care about the data at this point.
- We care *very much* about certain Ethernet headers:
  - Source MAC
  - Destination MAC
  - VLAN tag

Frame Size / MTU, still

- Pop quiz: What is the Standard Packet Size?
  - 576 bytes
  - 1280 bytes
  - 1452 bytes
  - 1492 bytes
  - 1500 bytes
  - 1518 bytes
  - 1522 bytes
  - 1540 bytes
  - 1542 bytes
  - 2304 bytes
  - 4352 bytes
  - 4464 bytes
  - 9000 bytes
  - *All* of the above? How?
  - ***None*** of the above? What?!?

Frame Size / MTU

- *Almost no-one means the same thing when they refer to "MTU" or packet size.*
- Even when referring to something seemingly-specific like "Ethernet Frame Size" you're likely wrong, and mis-identifying what you're talking about.
- Read Ethernet frame – Wikipedia and learn it well
- Now read Maximum transmission unit – Wikipedia and learn it well
- Finally, realize you didn't actually understand Ethernet Frame sizes *or* MTU at all!
- Your usage of the terminology will depend on what vendor's equipment you are interacting with, and that's normal
  - Just remember no-one speaks *exactly* the same language as you

MTU attacks? (Hanlon's razor, again)

- No. Just no.
  - If you think it's an MTU attack, it's because someone screwed up.
  - 100% of the time, all the time, every time.
  - Equipment reliably drops/ignores oversize packets.
  - Under-sized packets are normal and not generally cause for concern.
- However, *fragmentation* attacks have occurred and some routers/servers may still be vulnerable to those.

Data Link Layer – Ethernet Headers

- Every Ethernet (and Ethernet-like) device has a burned-in MAC address, called the "BIA", the "hardware address", or simply the "MAC address". This uniquely identifies every single computer on your network.
  - All modern OSes (mostly iOS & Android, but also others) can create fake MAC addresses on the fly. Apple and Google claim this is to "preserve your privacy", which in a school environment translates to "stop Admin from finding you". This is usually on by default on WiFi connections but *can also be enabled on wired connections*!
- Every Ethernet packet goes *from* a MAC address *to* a MAC address. This happens with or without IP, and is critical to troubleshooting many types of problems, even IP problems.

Data Link Layer – Ethernet Headers, cont'd

- The source MAC address will (ahem… *should!*) always be the MAC the sending device is using. There's no legitimate *end-user* scenario that sends packets with anything else in the Source field.
- The destination MAC address should only be one of three things:
  - Broadcast (ff:ff:ff:ff:ff:ff or ffff.ffff.ffff) – all devices on the subnet/VLAN must receive and process this packet.
    - Most common uses: ARP, older routing protocols, bored students == new red-teamers
  - Multicast (starts with "01:") – all devices on the subnet/VLAN receive and process this packet. Legitimately used by several protocols. Switches (including APs) should not forward certain specific types of these packets (e.g. BPDUs). **Modern NICs filter out most multicast traffic, only passing the "interesting" packets to the CPU.** See Multicast address – Wikipedia for more information.
    - Most common uses: Spanning Tree; CDP/LLDP/etc.; Flow control; LACP; 802.1X; and IP Multicast
  - Unicast (everything else) – only the destination device receives and processes the packet.

VLANs - overview

If you know how traditional VMs work, you've already got a pretty good idea how VLANs work:

- Virtual Local Area Networks (VLANs) allow you to emulate/multiplex/virtualize many "virtual" switches on a single hardware ethernet switch / single ethernet cable.
- Much like every VM has a namespace and usually a VMID or GUID to uniquely identify it, every VLAN has a VLAN ID (or VID).
  - In the IEEE 802.1Q standard (the only one we care about), VIDs range from 1 to 4096.
  - You don't get to use them all, there's always some reserved VIDs, which vary by vendor and product.
  - Most products let you attach names to VIDs, and when we refer to VLANs casually, it's often by that name, not the VID.
  - Only numeric VIDs, *not* names, are used when interconnecting multiple devices; switch #1 might call VID 1358 "OFFICE" while switch #2 calls VID 1358 "WAN". This can easily cause confusion if you don't have a way to keep them straight.
- To make VLANs work, a tiny (4-byte) VLAN ID field is inserted into the Ethernet header right behind the Source and Destination fields, just before the Data.
  - The TPID portion (first 2 bytes) is set to 0x8100, which is distinguishable from normal untagged Ethernet frames. The last 2 bytes encode the VLAN ID.
  - This increases the size of the Ethernet packet by 4 bytes, but switches mostly just take care of that detail for you, nowadays. You may need to know this when working with carriers, because some of them won't allow the extra 4 bytes. The real L2 hardware maximum MTU is at least 4 bytes larger on 802.1Q-capable switches than non-802.1Q switches.

Ethernet Header Attacks

- MAC address spoofing
  - Switches (and APs) have limited features to prevent this
    - Turning them on is usually a foot-gun deployment moment…
- MAC table flooding
  - Will never be “solved”, someone can always generate more fake MAC addresses than a switch can hold in its tables – it’s an asymmetric problem
- VLAN hopping attacks
  - Probably only theoretical for most networks
  - There are always rumours of zero-day vulnerabilities “on the wire”, but none are widely known, so ***if*** they exist, likely exclusive to nation-state or similar attackers
  - Attackable software bugs will always happen ☹

Hanlon's Razor

- 80%+ of the time, you (or someone else) have screwed up
  - Even with trivial attacks!
  - The other 19.9999999999% are script kiddies
    - I *said* they were trivial…
- Trust VLANs,
- Trust your switches, and
- Trust your routers.
  - Unless you're an EXTREMELY high-value target (i.e. top 0.0001%) in which case you should be running multiple dedicated switches on dedicated hardware inside an underground vault anyway… this talk isn't for you!
  - Or until proven guilty, anyway – you have to start *somewhere!*
- BUT: don't assume that your network engineers know what they're doing, and also don't blindly trust your vendors.

Layer 3 - Network

- ARP
- IP

- Many others exist, but are nowhere near as common.
  - Long-tail is a good place to look for exploits…

ARP: The glue between the layers

- IPv4 addresses map directly to Ethernet MAC addresses…
  - …somehow.
  - Address Resolution Protocol (ARP) runs on every broadcast-capable medium, which usually means both Ethernet (and ether-like) and WiFi. (Other types exist, but are increasingly rare.)
  - See Address Resolution Protocol – Wikipedia for all the gory details.
- ARP is not IP! An ARP packet is *not* just a type of IP packet!
  - ARP is part of the Internet Protocol *suite*, but is a separate, parallel L3 protocol that runs *alongside* IPv4.
  - IPv6 does not use ARP, it uses NDP instead. See Neighbor Discovery Protocol – Wikipedia if you need more details.

Should I *ever* block or filter ARP?

NO!

- You'll create an incredibly fragile system that won't survive the first hardware replacement, that also won't permit any troubleshooting.
- The entire IP stack assumes that ARP "just works". See previous comments about the internet still working mostly on convention.

How ARP works, briefly:

- Host A needs to send an IPv4 packet to Host B.
- Host A realizes it doesn't know Host B's MAC address, and therefore can't yet send an Ethernet packet there.
- Host A checks to see if Host B is on the same subnet as Host A, and if so:
  - Host A sends a broadcast Ethernet packet containing an ARP Request (or "who has") message inquiring about a specific IP address.
  - Host B receives the broadcast message, realizes "this is for me!" and replies *directly* back to Host A (since A's MAC address is conveniently available from the Request packet's Ethernet Source Address header) with an ARP Response (or "is at") packet.
  - Host A receives the ARP Response, and updates its internal cache of IPv4-to-Ethernet address mappings (unsurprisingly called the "ARP table").
  - Host A now knows Host B's MAC address and can address the Ethernet packets to Host B correctly.
  - Host A now sends the original IPv4 packet to Host B's MAC address.
- If Hosts A and B are on different subnets, IP routing is used instead – but ARP still plays a part, as we'll see shortly.

ARP Attacks

- Spoofing
  - Anyone can send an arbitrary packet onto the network
- Resource Exhaustion
  - Hosts' ARP tables are not infinitely large
- …that's about it

Hanlon's Razor

- Almost all ARP misbehaviour is because someone screwed up:
  - Duplicate IP addresses
  - Incorrect subnet masks
  - Wrong VLAN ID
- If you have neglected physical security, network segmentation, and all forms of L1/L2 access controls, then:
  - OK, yeah, it could be an attack
- Note: Insider attacks are trivial to execute
  - No good technical defenses other than good security hygiene at all the layers
  - Do not hard-code ARP entries as a defense: your successor will hunt you down and ensure you die a slow, painful death (fragile, not resilient, prevents DHCP, etc.)
  - ***Advanced troubleshooting is indistinguishable from an attack***

ICMP – the internet's control protocol

- ICMP was designed to do anything and everything
- We realistically only have ~5 use cases today (out of 20+)
  - Echo Request/Reply (ping)
  - TTL Exceeded (traceroute)
  - Destination Unreachable
  - Fragmentation Needed – critical to the modern internet
  - Source Quench – rarely seen but still needed
- Like ARP, part of the IP suite
- Unlike ARP, ICMP packets are regular IP packets

ICMP Blocking 1/3

- From http://shouldiblockicmp.com/ :

No!!

- Scrub? Yes
- Rate-limit? Yes
- Block outright? No

ICMP Blocking 2/3

- For → prevents discovery
  - If your host is online in any other way, it's already discoverable
  - "Security by Obscurity" is a serious anti-pattern in modern security practice
  - Horizontal probing can use ARP instead
- Against → breaks major functionality
  - Timeouts are now *all* mystery problems
  - You can no longer troubleshoot most classes of problems
- No-one *ever* unblocks ICMP to let you troubleshoot
  - This needs to be prevented on day zero
- The mythical/obsolete security benefits are outweighed by operational considerations
  - Do you want me to be able to *fix* your ultra-secure network when something breaks, or not?

ICMP Blocking 3/3

- DO NOT:
  - Make every IP behind the firewall pingable from the internet for no good reason
  - Make every PC pingable
- DO:
  - When you allow a protocol through the firewall/ACLs, include ICMP
    - Reduces customer complaints: “it’s down!” != “I can’t ping it”
  - Make L3 switches, routers, and servers pingable, on the inside network
    - So you can tell if your own network is up or not
    - ICMP is handled much lower down in the stack than an HTTP request, and will let you distinguish between a broken VM/NIC/OS and a broken webapp
  - Scrub incoming ICMP at the firewall

ICMPv6 Filtering

- This one is easier: ***just say no!***
- Filtering ICMPv6 completely breaks IPv6!

ICMP Attacks

- …definitely exist
  - But most are obsolete
    - All your servers are fully patched, right? Right??
  - The era of serious ICMP attacks was ca. 1990-2005 C.E.
- What about reconnaissance?
  - ICMP makes reconnaissance slightly easier
  - Reconnaissance can still be done without ICMP
- ICMPv6 is sure to be a Brave New World of Vulnerabilities
  - …but you have no choice but to allow it
  - No-one really knows yet if you can safely filter by type

Hanlon's Razor

- Most ICMP is normal & natural
  - E.g. client systems ping their printers regularly "just because they can"
  - Many routers, load balancers, app firewalls use ICMP under the hood to *continuously* detect reachability
  - Lots of people use ICMP to self-diagnose before calling support
- Attacks (mostly obsolete and harmless) occur all the time
  - May as well scrub those if you can
- Rate-limit ICMP (even internally) to prevent carpet-bombing
- Filter ICMP types (even internally) to prevent corner-case exploits

Layer 4 - Transport

- Internet Protocol (IP)

- Others exist, but who cares about them?

IP – finally!

- Packet header contains 3 critical parts:
  - Source IP address
  - Destination IP address
  - Protocol

- Remember:
  - Every Layer 3 IP packet is **also** a Layer 2 802.2/Ethernet-II packet with MAC addresses
    - Every layer 2 802.2/EthernetII packet is **also** a Layer 1 802.3/Ethernet packet with preambles, etc.
      - Every layer 1 802.3/Ethernet packet is **also** a series of physical impulses in a communication channel
        - Every physical impulse is **also** a series of atomic/subatomic events… *cough*
          - OK, we rarely need to care about L1 or lower unless it breaks
            - If atomic force breaks we're all screwed anyway

- …L4+ is actually the most boring part of the stack, not L3

So boring…

- It’s so boring I have relatively little to say
- We’re almost done

IP routing

- From the ARP slides:
  - Host A checks to see if Host B is on the same subnet as Host A, and if so…
  - If Hosts A and B are on different subnets, IP routing is used instead – but ARP still plays a part, as we'll see shortly.
- If Hosts A and B are on different subnets:
  - Host A picks the most appropriate router
  - Host A ARPs for that router
  - Host A sends the packet to that router's L2 MAC address
  - Lather, rinse & repeat until you get to a router that has an interface in the same subnet (i.e. it can use ARP to resolve the final destination L2 MAC address)
  - Final router sends the packet to the destination's L2 MAC address

IP routing 2

- IP routes are expressed as IPADDR/MASK, or A.B.C.D/X
  - E.g. 1.1.1.0/24
- Let's say we're going to Cloudflare (1.1.1.1)
  - ```
    Route:          00000001000000010000000100000000
    AND MASK:  AND 11111111111111111111111100000000
    AND DEST:  AND 00000001000000010000000100000001
    gives us:    = 00000001000000010000000100000000
    ```
- IF <ouraddr> AND <mask> == <dest> AND <mask>
- THEN host is on same subnet, use ARP
- ELSE host is on different subnet, look for route
  - IF <route> AND <mask> == <dest> AND <mask>
  - THEN that route is valid for <dest>
    ```

IP routing 3

- When a host or router looks for a matching route, it always looks for the LONGEST MASK that produces a match
- So if I have routes for
  - 1.1.1.0/24
  - 1.1.1.0/28
  - 1.1.1.1/32
- …the longest matching route (1.1.1.1/32) will be used
- That's 90% of IP routing.

IP routing weirdness

- IP Redirects
- Conflicting routes on different devices
- Accidentally-overlapping routes

- On the local scale, attacks are almost always indistinguishable from incompetence
- Firewalls can scrub both IP header malice *and* idiocy, so use them to do that

Hanlon's Razor

- Repating from the last slide:
  - On the local scale, attacks are almost always indistinguishable from incompetence
- You all know that ~99.999999999…% of the traffic on a network is normal and legitimate
- Rely on Firewalls/IDSes/IDPs to find the remainder

IP Routing - BGP

- Out of scope for this talk
- Lots of real-world attacks
- See https://manrs.org/ to start learning about routing security
- RPKI and IRR are starting to make a difference globally

L4 - Transport

- In 2024, this is **TCP** and **UDP** and … more or less nothing else
- Both protocols add:
  - Source port
  - Destination port
- At the stack/OS level, the port is used to differentiate between applications running on the same host
- This is where most of you start worrying about security!

L4 Attacks

Yes

- There’s not much more to say about it!
- Most random L4 ports are, in fact, probes
- Easy enough to detect legit traffic – it’s on the port it should be on
  - Some attack traffic comes in on the legit port too, that’s a Layer5-7 attack, which is out of scope for this talk

Hanlon's Razor

- Here we have a dilemma
  - The off-port traffic is clearly not legit
  - Enough illegitimate traffic comes in the legitimate port to be an issue
- Application Firewalls try to solve this problem
  - Mediocre solution, but better than nothing

- It could be legit, it could be an attack
  - Figuring this out is hard

Layers 5-7

- Not my problem today
- Listen to the other speakers this weekend!!!

Layer 8

- For completeness only:
  - Layer 8 is not part of the OSI stack
  - Layer 8 (roughly) refers to business, financial, or political decisions that are the source of problems in Layers 1-7

The End

- That’s all, folks!