# Network Security 6: Network Elements

i.g.batten@bham.ac.uk

# Spectrum of boxes

- Hubs
- Simple Bridges
- Switches
- Filtering Switches
- Routers
- Filtering Routers/Firewalls (simple firewalls)
- Stateful Firewalls (mainstream firewalls)
- Deep Packet Inspecting Firewall
- (IDS/IPS/spooky things)

# Hubs

- Copy ethernet packets from one interface to all interfaces, probably just electrically (ie, don’t look at the packets at all).
- Provide no security, and are (let’s hope!) being taken out of networks.

# Simple Bridges

- Copy ethernet packets from one interface to the others, dropping packets that are malformed or corrupt.
- Provide almost no protection, aside from extremely simplistic flooding attacks not seen since the 1980s.

# Filtering Bridges and Switches

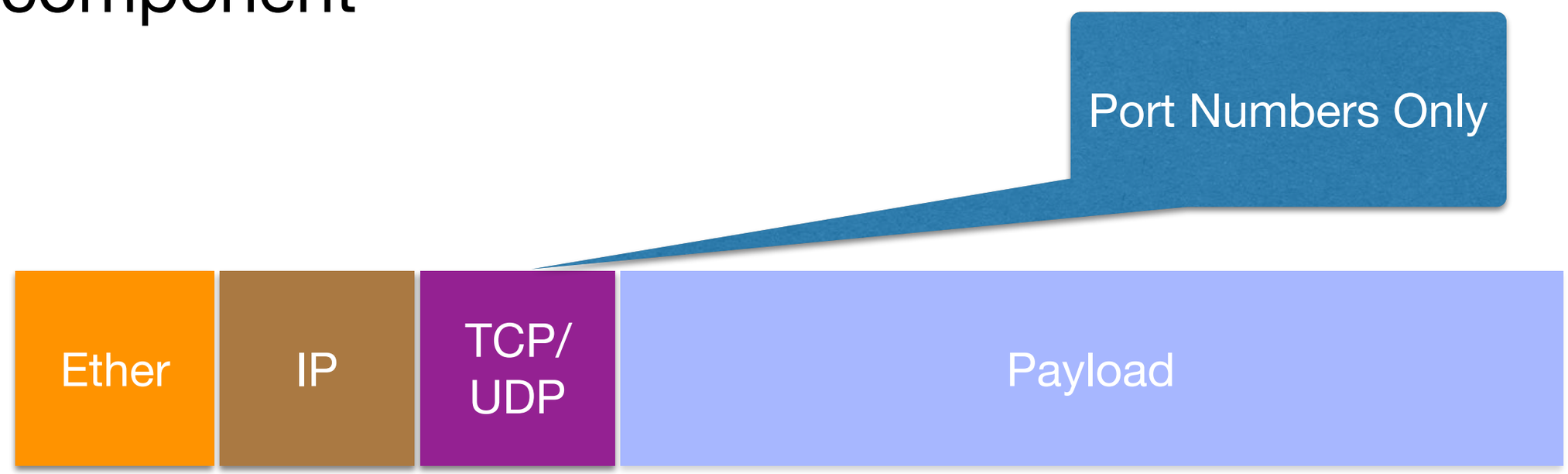
- Copy ethernet packets from one interface to the other, if the bridge believes that the source and destination are on different sides of the bridge.
- Prevents an attacker on network A from observing traffic between two hosts on network B.
- Switches stop hosts from seeing anyone else's traffic (although not hard to bypass)

# Routers

- Looks at the IP header and chooses an interface to send the packet out of
- Blocks some malformed packets, and has similar filtering effects to a switch.
- Doesn’t (in general) propagate broadcast packets

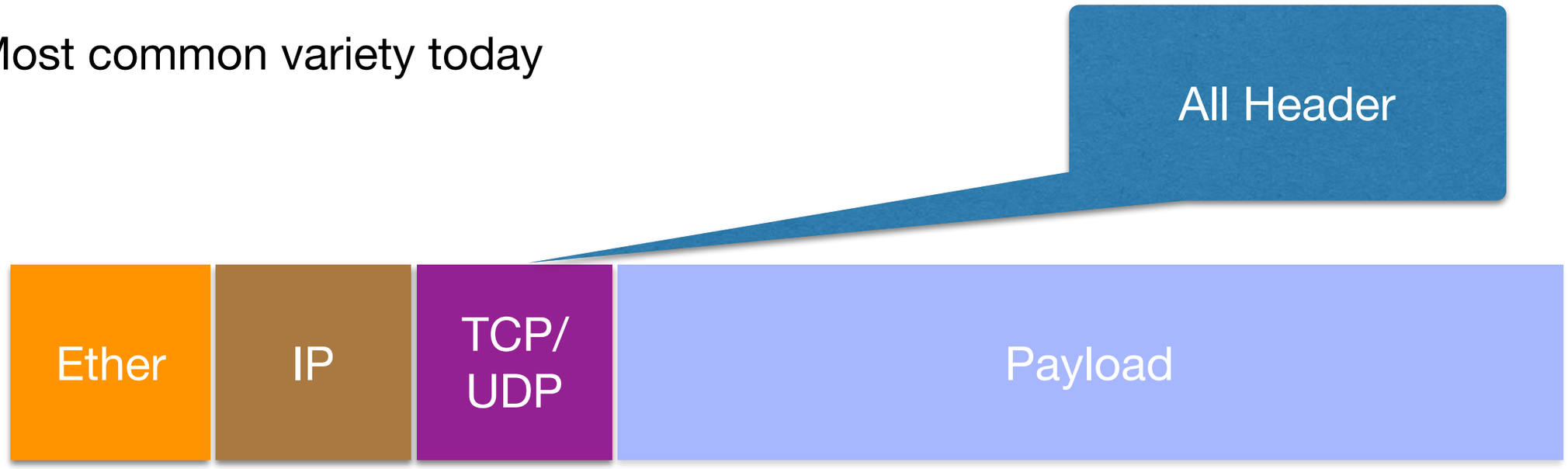
# Simple Firewalls

- Act as a router, but also look at TCP and UDP port numbers and block/pass based on those values
- Will shield services on systems from an attacker
- The minimum requirement to be a security component



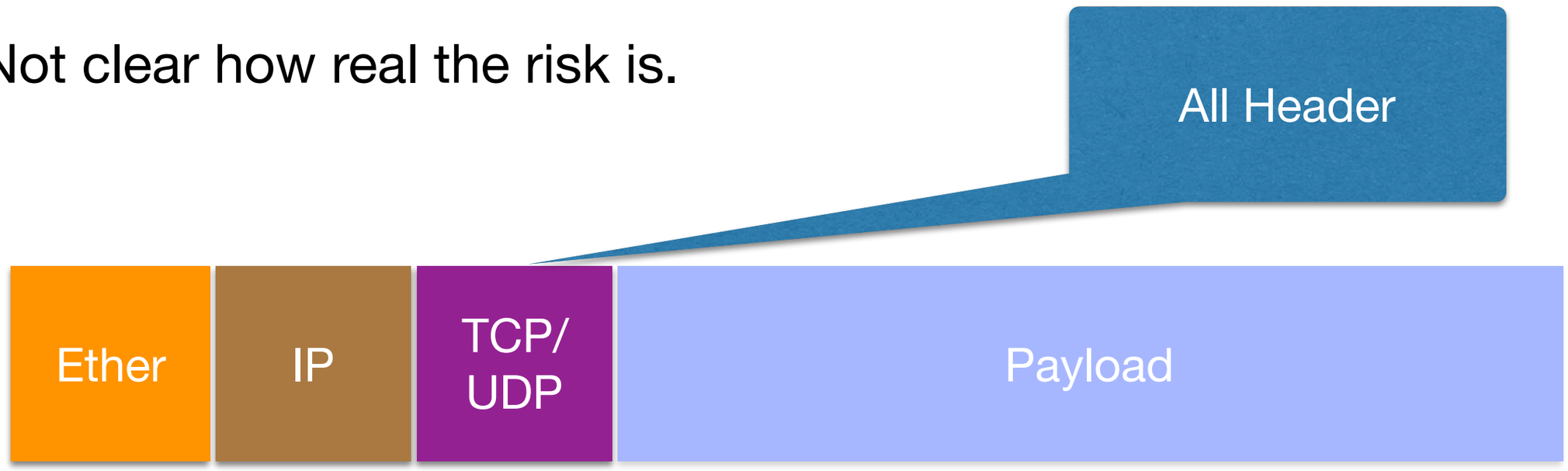
# Stateful Firewalls

- Look not only at port numbers, but the whole state of the TCP connections through a router.
  - Tracks the connections, so it knows that SYN-ACK or RST are the only sensible responses to SYN
  - Tracks sequence numbers, to ensure the next packet is actually a sensible packet for the connection
- Not only shields services, but blocks more complex attacks on TCP connections (next week!)
- Most common variety today



# Stateful Firewalls++

- There are some firewalls (Cisco ASA, for example, and the FreeBSD pf suite) which can rewrite TCP sequence numbers, reassemble fragments and re-order packets.
- Attacks on sequence numbers will come next week
- Idea is that protected systems only ever execute the most well tested code paths, so attacker cannot probe weakness in less well tested code.  Makes all TCP stream canonical.
- Not clear how real the risk is.



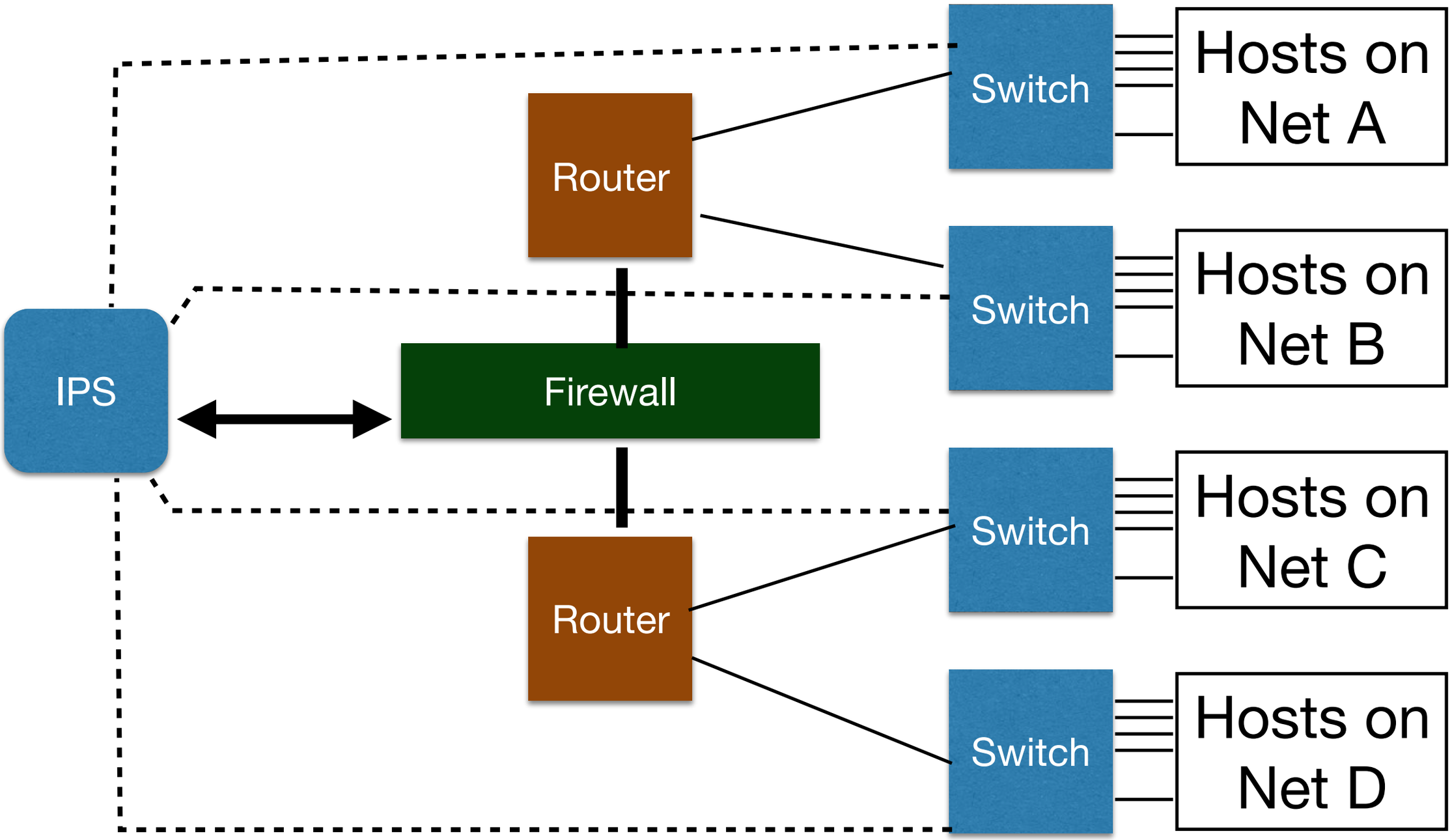
# Deep Packet Inspection

- Looks at entire packet as it passes through router
- Can operate as a virus scanner, can check protocols aren't being abused, can perform arbitrary re-writes.
- Can even block encryption it doesn't know about by considering randomness, although that is an arms race.
- Sometimes regarded as unethical when deployed in company networks, arguably regulated in public networks (Phorm debacle).

# IDS/IPS

- Intrusion Detection / Prevention Systems
- Receives copy of traffic from switches and routers
- Examines whole packet to looks for attacks and other worrying conditions
- Either reports them (D) or instructs routers and firewalls to drop traffic (P).

# IDS/IPS


Router
Switch
Hosts on Net A
Switch
Hosts on Net B
IPS
Firewall
Router
Switch
Hosts on Net C
Switch
Hosts on Net D

# So what can we do with the components?

- We want to detect threats,
- block threats,
- respond to threats

# Switches and Bridges

- MAC-based filtering is about local traffic, as all remote packets will have local MAC addresses from the last router they went through
- However, can be useful to stop attacks from insiders, and to avoid common configuration mistakes.

# Switch filtering

- Although it is very labour intensive, access ports can be filtered to only accept traffic from the expected device.
  - Prevents MAC spoofing and addition of cascaded switches and hubs to add unauthorised devices
- Doesn’t scale well
- Also worth doing in the vicinity of routers to make “whoops, I plugged the wrong cable” errors less fatal

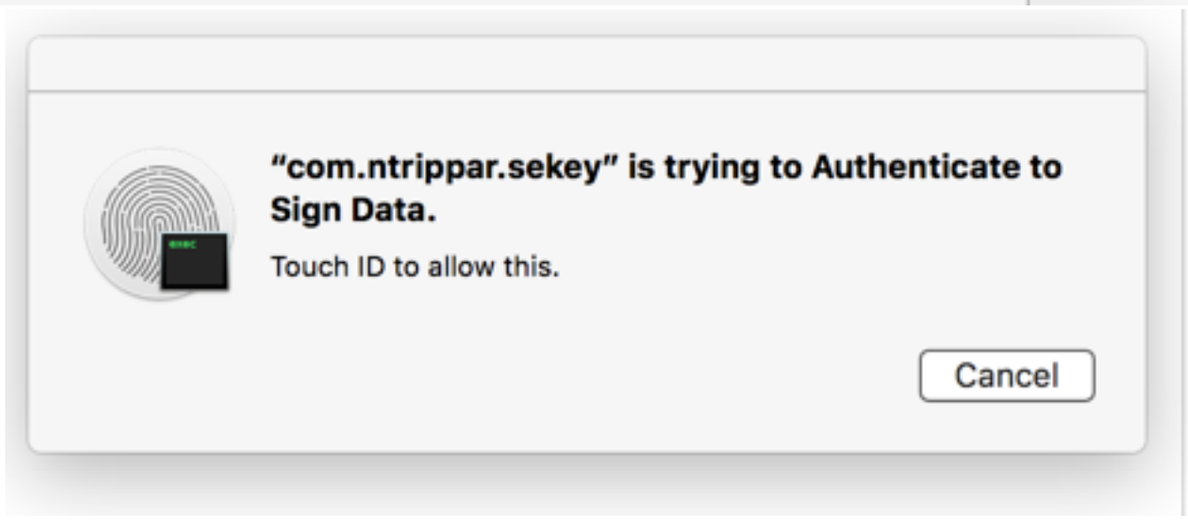
# Switch Authentication

- You can automate switch filtering with 802.1x authentication: devices are authenticated to the switch before they can send traffic, somewhat in the manner of wireless authentication.
  - Also “posture checking” on things like virus scanners and patch levels
- Very fashionable in about 2005–10, lots of kit sold, but not clear how many deployments succeeded.

# Problems with 802.1x

- Basic idea is sound: when a port sees a new device, it requests authentication before enabling the port, passing the MAC address, allowing the MAC address onto a designated VLAN, etc.
- Problems arise with:
  - Cascaded switches (which one do you authenticate to?) (“multiple supplicant”)
  - Failure of radius/etc server is potentially devastating
  - Printers and other “non-user” devices: where do the keys go?  Securely?  TPM offered hope, as does Trustzone/SGX/SEP but little enthusiasm from suppliers.
  - Machines that need to boot unattended, ditto.
  - If doing posture analysis, do you really have the agent for all your equipment?  No?  So how do you know which devices should be running it and which can’t.
  - If he’s here, Darren will now tell us his war stories…

# In passing, SEP…


# Routers

- Prevent attacks involving sending packets that are not IP: routers will not pass ARP, weird non-IP protocols (assuming router is not configured to route them), etc.
- On their own, routers do not provide any filtering

# Simple Firewalls

- Can be part of a router or part of an end system (and a “router” may be a computer with two interfaces).
- Conceptually usually though of as being “in front” of the network interface, but in reality is usually part of the first code run when a packet arrives (for ingress filtering) or the last code run when a packet leaves (for egress filtering) (more complex on Linux)

# Simple Firewalls

- Allows the blocking of everything apart from certain protocols and ports, or…
- …the passing of everything except certain protocols and ports.
- Cannot check legality of TCP packets at more than the most basic level (packets in isolation, rather than as part of a flow)
- Struggles to filter and check UDP.
- Struggles in the face of NAT
- Unless you are using an ancient Cisco router, unlikely to be used for any security application today

# Stateful Firewalls

- Inspects all the elements in the packet header, and keeps track of all connections through the firewall
- Also called, unsurprisingly, “connection tracking” (also “keep state”, from the configuration option in the BSD firewall code).
- Code is closely related to NAT (my memory is that it predated NAT, which leveraged the code, but I could be wrong).
- Can check that TCP packets are in the expected state, as we will explore next week.
- Match UDP packets heading out and coming back in, to permit services to cross the boundary.
- Performance implications in complex or busy networks, and require careful engineering to avoid intentional and unintentional denial of service attacks.
- Failover and standby configurations notably tricky, especially in the face of asymmetric routing
- Most common form of firewall today

# Stateful++

- (This is my terminology).
- As well as checking that packets are expected, modifies them to make them “more” expected: removes duplicates, waits for retransmissions, reassembles fragments.
- Protects against unknown risks in rare code paths; packets only arrive in common states and conditions
- Also means that hardware assist and “fast path” is always used, with rare/untested “slow path” not needed
  - You have to worry about the test coverage in TCP stacks.
- Understands HTTP, SMTP, FTP and so on and similarly enforces “sensible” rather than “strict” limits on command length, syntax, etc.
- Basic idea is that your ASA (or whatever) is better tested and less sensitive than your servers.

# A recent disaster

- gethostbyname (“0124023429832498573298”) could, for a suitably large string, provide arbitrary code execution.
- Long-running (2001–15, at least) bug in GNU libc, with some pointer arithmetic not being done correctly
- You’ve got a problem.  You use pointer arithmetic.  Now you’ve got two problems.

# Use as remote attack

- SMTP mail protocol starts with the sender introducing themselves with “HELO my.host.name”.
- Protocol pre-dates DNS and this was only way to pass the name; in 1980 everyone was trusted to tell the truth.
- my.host.name is not normally checked (we just do a DNS reverse lookup on the source IP number these days) but some SMTP servers do use it as part of spam checking.
- Remote attack involves sending:
- HELO 0124023429832498573298{and longer}\r\n
- Buffer overrun / stack smash / etc

# DPI stops this

- DPI firewalls boxes look at SMTP sessions and perform some basic sanity checking, and only pass “sensible” instructions to the mail server.
- Prevent attacks involving weird character sets, buffer overruns, etc.
  - Cisco ASA limits all SMTP commands to 80 characters, while this attack requires at least 4096.
- However, in 2018, an awful lot of email is server-to-server encrypted (SMTP+TLS) so these defences don’t work.  Decrypting the session is a risk/return decision.

# IPS/IDS

- Scans all packets on the network for problems
  - Mostly ancient exploits with static patterns, but on networks with poor patching they are still serious.
  - Might maintain state about connections too
  - Looks for trouble and shuts it down by either dropping packets (“in line”) or by instructing firewalls to add dynamic rules
  - In many cases, false positive rate renders it unusable.
  - Again, it would be interesting to see how many deployments are being used seriously.