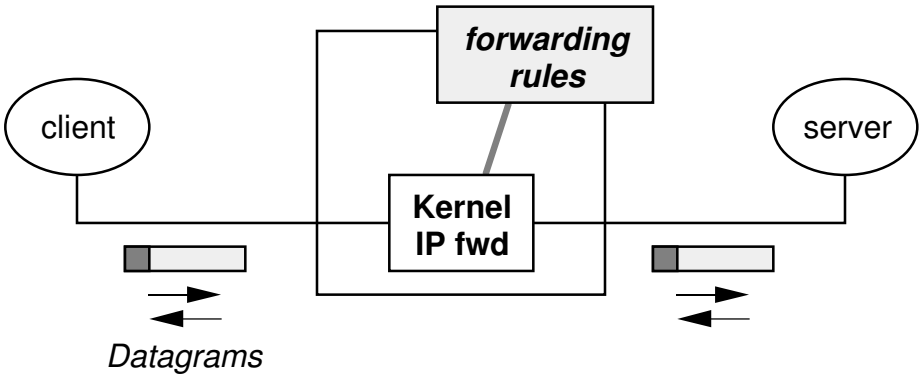
NETWORK SECURITY

- **On a big network we can't rely just on host security**
  - *have you secured* **every** *service on* **every** *host?*
  - *how do you know nobody has brought in an insecure host and plugged it in themselves?*
- **We need to block "insecure" services from outside access**
- **We want a single point of access control and logging**

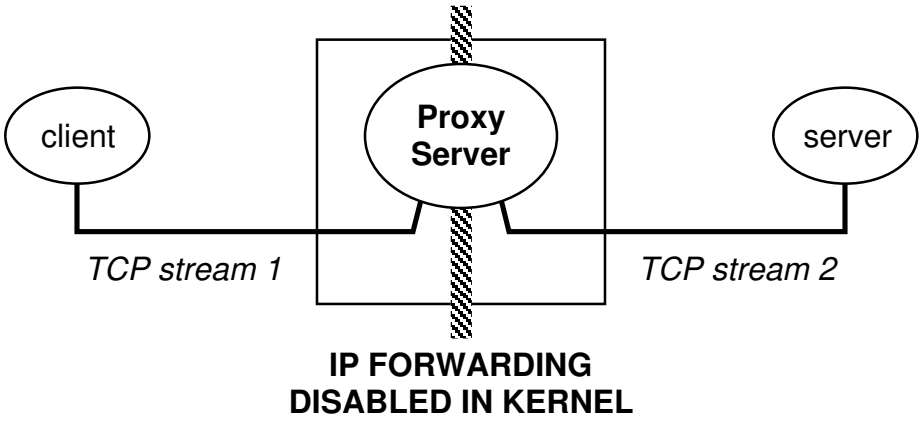
A FIREWALL

Firewalls come in many flavours…

- **PACKET FILTER** checks datagrams against pre-set rules to decide whether to forward or discard them



- **PROXY SERVER** contacts an external server on behalf of an internal client (so we can block all other in-out traffic)

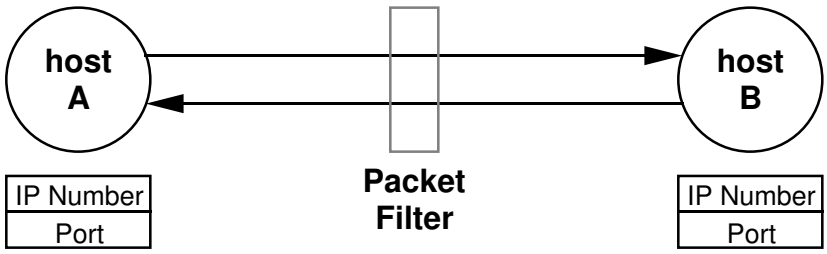


Firewalls are notoriously difficult to set up properly

- **A bad firewall is worse than no firewall at all**
- **Very easy to leave security holes without realising it**
- **How secure is your firewall machine itself?**

PACKET FILTERS

(a.k.a. "Access List" on Cisco)



- **Filter has a list of rules**
- **Rules are tested in order – first match is used**

  Accept: Forward the datagram

  Deny: Discard it

  Reject: Discard it and send back ICMP "host unreachable"

- **Transparent to clients**

Two approaches to policy:

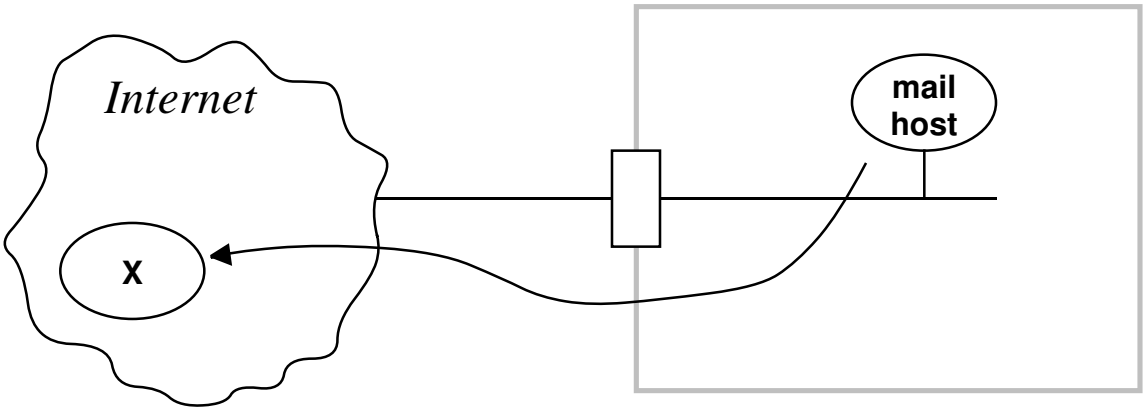
- Leave open and block "insecure" services

  – which ones? have you got them all?

- Block everything, and open only services which you know are both secure and needed

PACKET FILTERS (2)

- **Example: enable outgoing SMTP from mail host**



Any IP
Port 25

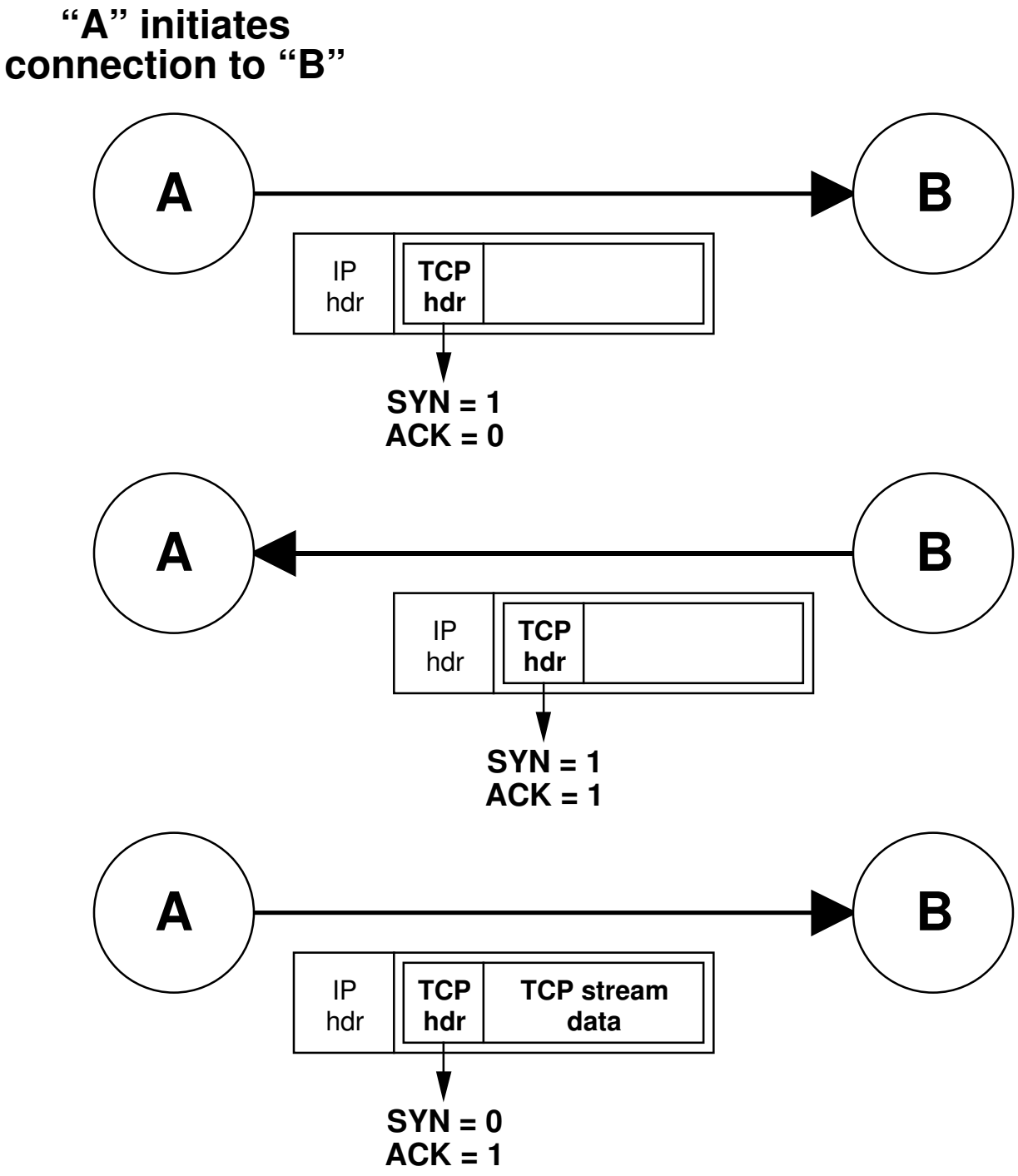
mail host
Any Port

	Protocol	Src IP	Src Port	Dest IP	Dest Port	Action
1.	**TCP**	**mail host**	**Any**	**Any**	**25**	**ALLOW**
2.	**TCP**	**Any**	**25**	**mail host**	**Any**	**ALLOW**
3.	**Any**	**Any**	**Any**	**Any**	**Any**	**DENY**

BEWARE!!! Either end can open the TCP connection

You have just permitted connections from port 25 on **any** Internet host, to **any** port on your mail server!

TCP three-way handshake



All subsequent segments have SYN=0, ACK=1

FIREWALL EXAMPLE

Permit ONLY outgoing SMTP from our mail host to any machine on the Internet

Rule 1. FROM outside [any IP, port 25]
TO inside [mailhost, any port]
AND SYN=1, ACK=0

DENY

Rule 2. FROM outside [any IP, port 25]
TO inside [mailhost, any port]

ALLOW

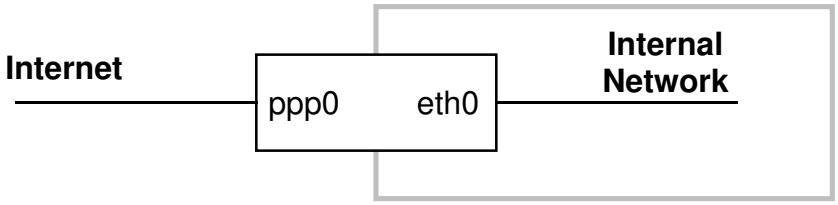
Rule 3. FROM inside [mailhost, any port]
TO outside [any IP, port 25]

ALLOW

Rule 4. DENY EVERYTHING

Note: the order of rules is critical
(Packets matched by rule 1 also match rule 2; the FIRST match is used)

More Packet Filter Examples



Disable all telnet through firewall

DENY tcp from (any IP, any port) to (any IP, port 23)

Disable *incoming* telnet only

DENY tcp from (any IP, any port) to (any IP, port 23) incoming via ppp0, and SYN=1/ACK=0.

Prevent IP spoofing

- Datagrams from the Internet should never have a source IP number = one of our internal IP numbers
- Filter them out (and log whenever they occur – probably an attack)
- We can also protect the Internet from spoofers inside our net!

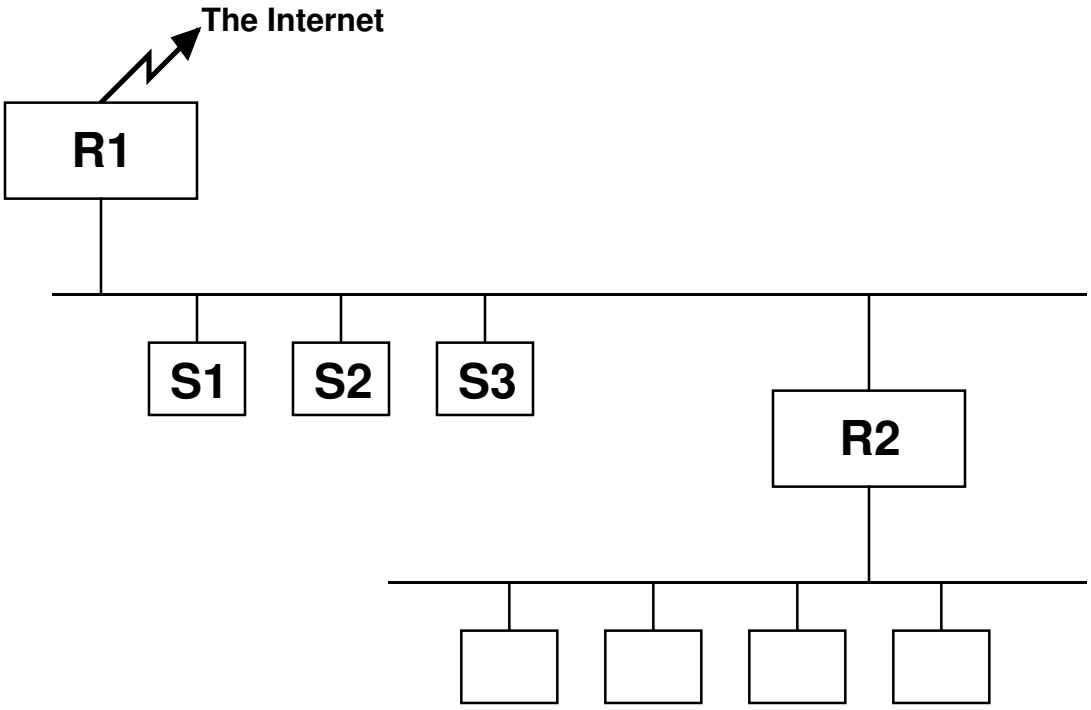
UDP is a problem for firewalls

- No way to tell who initiated the exchange (no SYN/ACK bits)
- Many insecure protocols are UDP-based (e.g. NFS)
- So many sites just block UDP altogether
  – *apart from queries between DNS servers (source and dest port 53)*
- **traceroute** sends UDP probes, so you can't traceroute through these firewalls

Fragments are a problem too

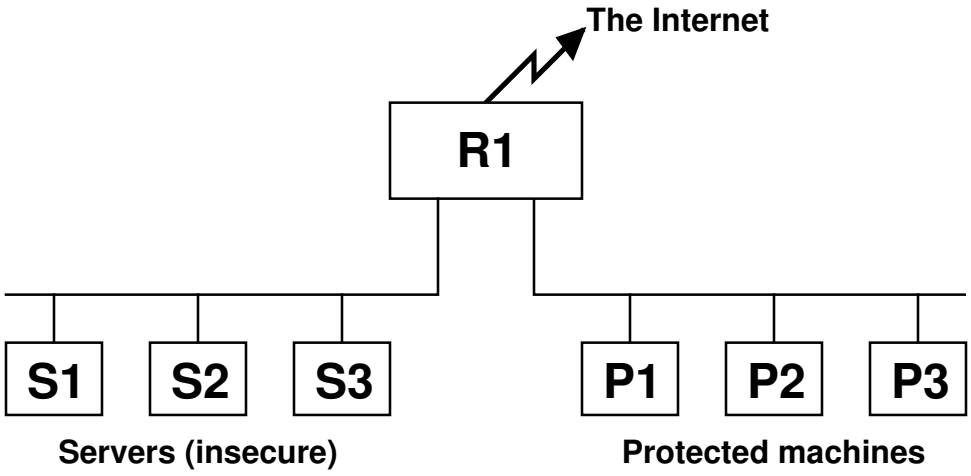
- TCP and UDP port numbers are only in first fragment
- Check the first fragment, let all the others through regardless
  – *vulnerable to "fragment bomb" attacks*

FIREWALL ARCHITECTURES: "Screened Subnet"



- Perimeter (Red) network has our servers accessible to outside: web, ftp, mail etc.
- Internal (Blue) network has our secure systems
- R2 is configured to block access from S1...S3 to Blue network – so even if hackers break our servers, they can't get in any further

FIREWALL ARCHITECTURES: Router with dual interfaces



R1 packet filter configured to:

- ✓ ALLOW from P1...P3 to Internet
- ✓ ALLOW from P1...P3 to S1...S3
- ✓ ALLOW from Internet to S1...S3
- ✗ BLOCK from Internet to P1...P3
- ✗ BLOCK from S1...S3 to P1...P3

Functionally equivalent to screened subnet – but one less router needed

Linux packet filtering

- ***Three sets of firewall rules:***

  -I Input
  -O Output
  -F Forwarding

  see **man 4 ipfw**

  ipfwadm -Fln — *Forwarding, list rules, numeric display*

  ipfwadm -F -f — *Forwarding, flush all rules*

- ***Set policy (default if no rules match)***

  ipfwadm -F -p **accept** | **deny** | **reject**

- ***Add new rule to end***

  ipfwadm -F -a **accept** | **deny** | **reject** [**-P** **tcp** | **udp** | **icmp** | **all**] [**-W** *eth0*]

  Protocol *Interface*

  -S *x.x.x.x/n* [*port[:port]*] *Source*

  -D *x.x.x.x/n* [*port[:port]*] *Destination*

  [**-y**] *Match only SYN=1 ACK=0*
  [**-k**] *Match only ACK=1*
  [**-b**] *Bidirectional rule*
  [**-o**] *Log if rule matches*