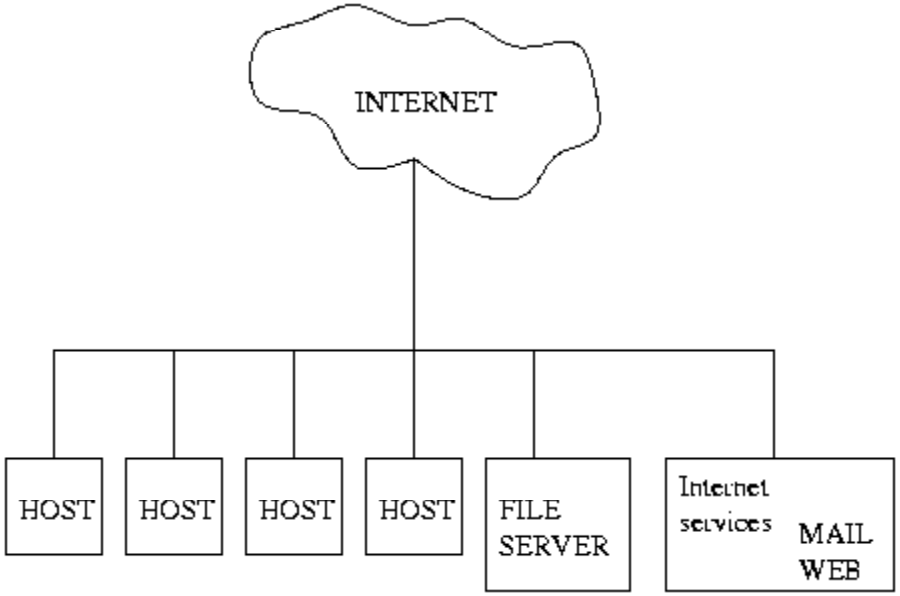
Internet Firewalls and Network Security

In days of old, brick walls were built between buildings in apartment complexes so that if a fire broke out, it would not spread from one building to another. Quite naturally, the walls were called ”firewalls”.

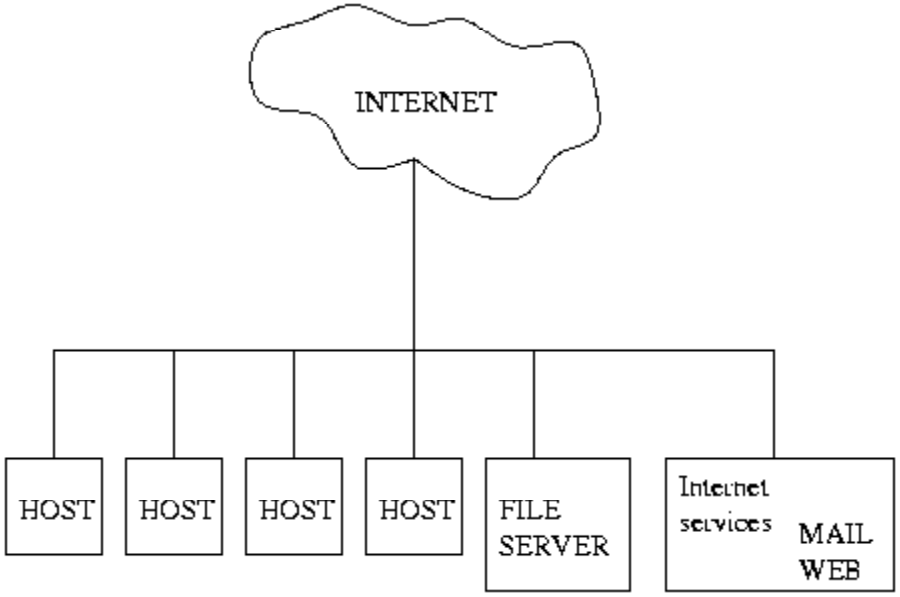
Table of Contents

The Unprotected Network



What could possibly be wrong with this setup?

The Unprotected Network



Hackers paradise & administrators nightmare!

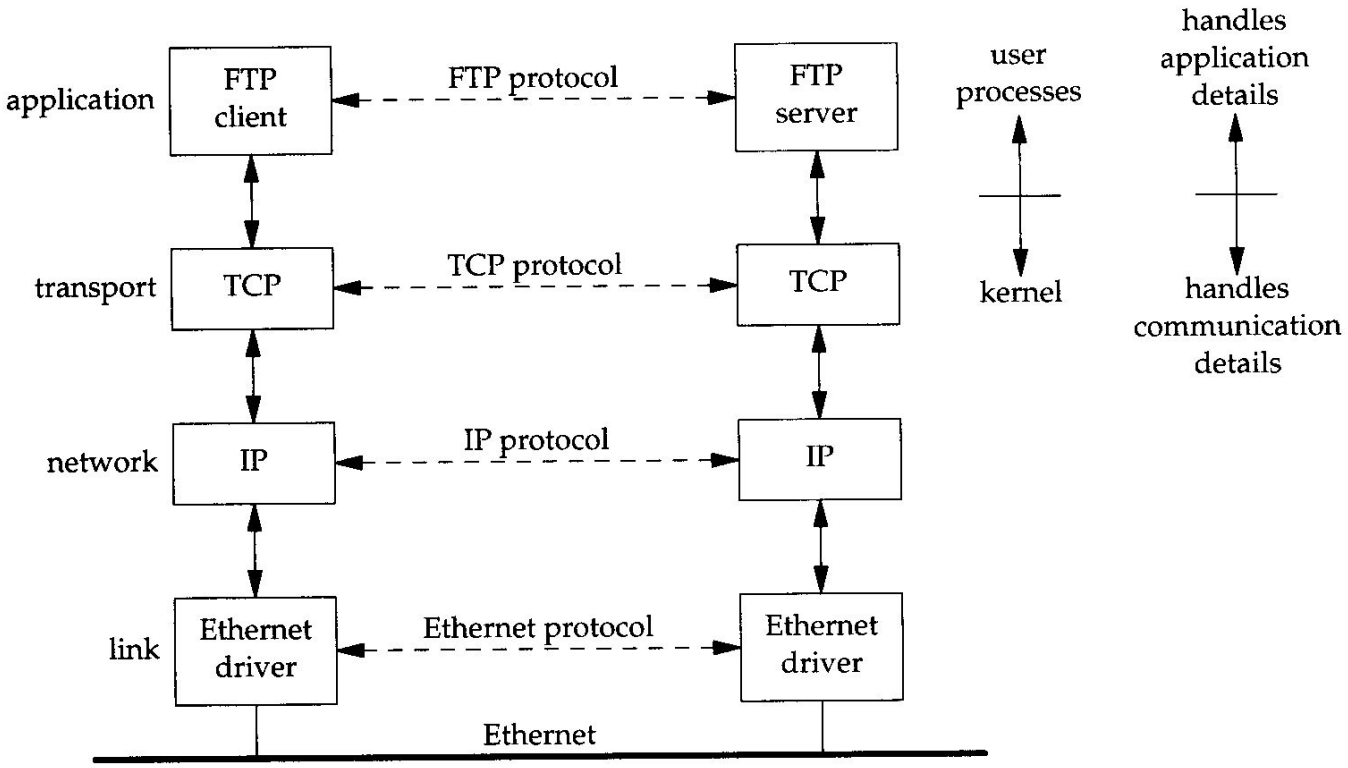
What Can We Do?

Fortunately firewalls can give us very good protection against attacks from the Internet.

The only problem is that there are numerous firewall strategies.

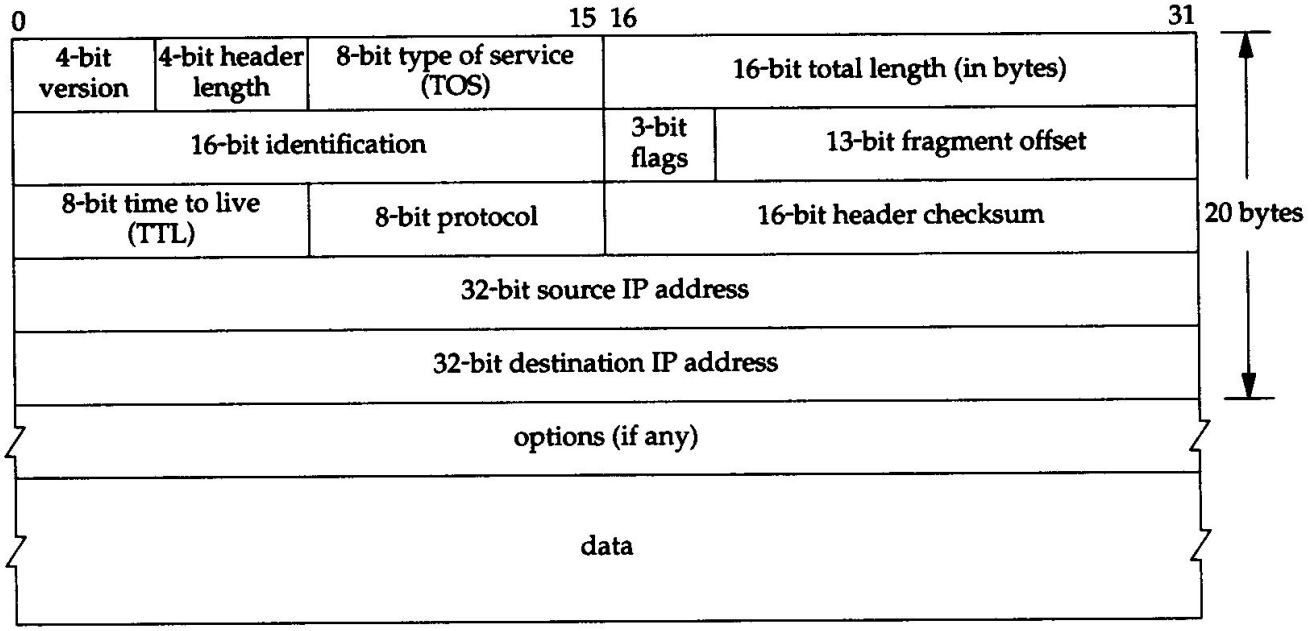
In order to choose the right strategy we need to know a bit more about the underlying communication protocol TCP/IP.

The IP-stack


application
FTP client
FTP protocol
FTP server
transport
TCP
TCP protocol
TCP
network
IP
IP protocol
IP
link
Ethernet driver
Ethernet protocol
Ethernet driver
Ethernet
user processes
kernel
handles application details
handles communication details

The IP-protocol

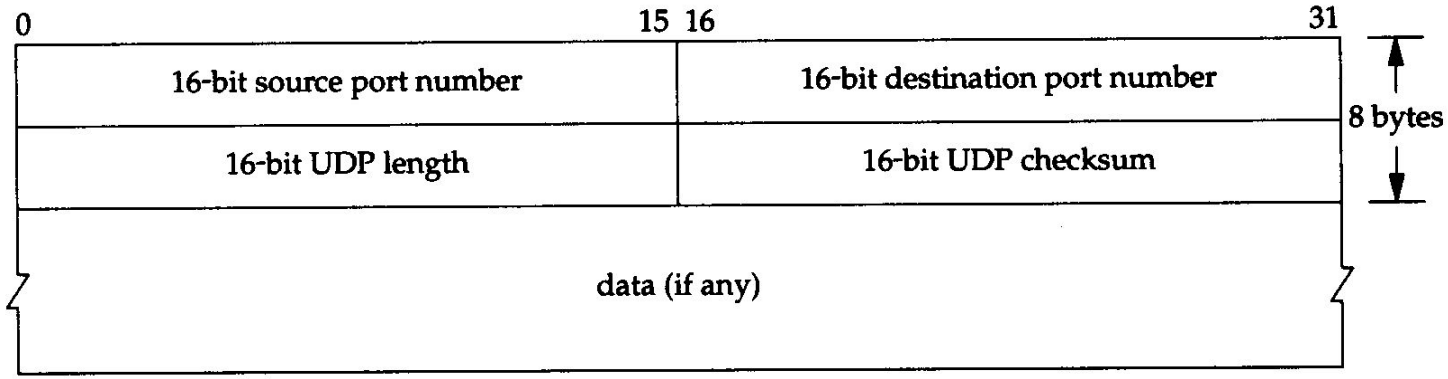
IP Header



Interesting fields: source and destination address

The UDP-protocol

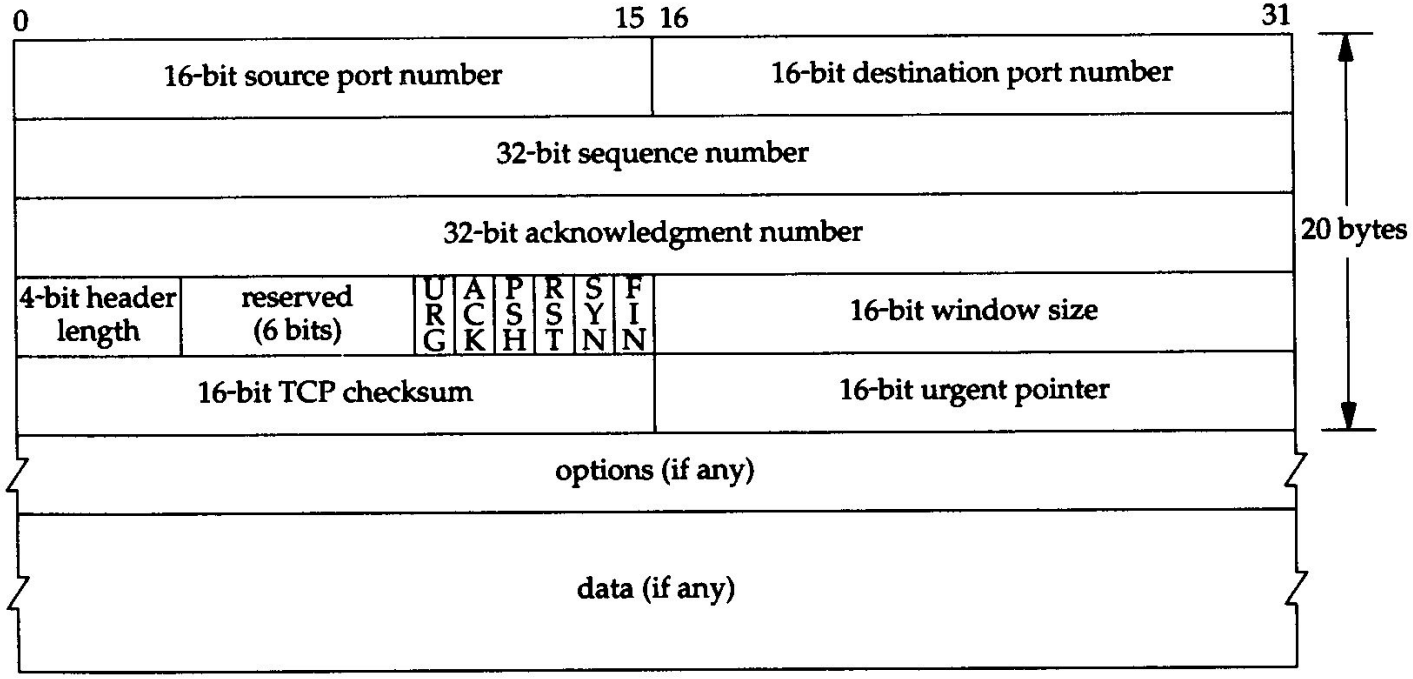
UDP Header



Interesting fields: source and destination port

The TCP-protocol

TCP Header



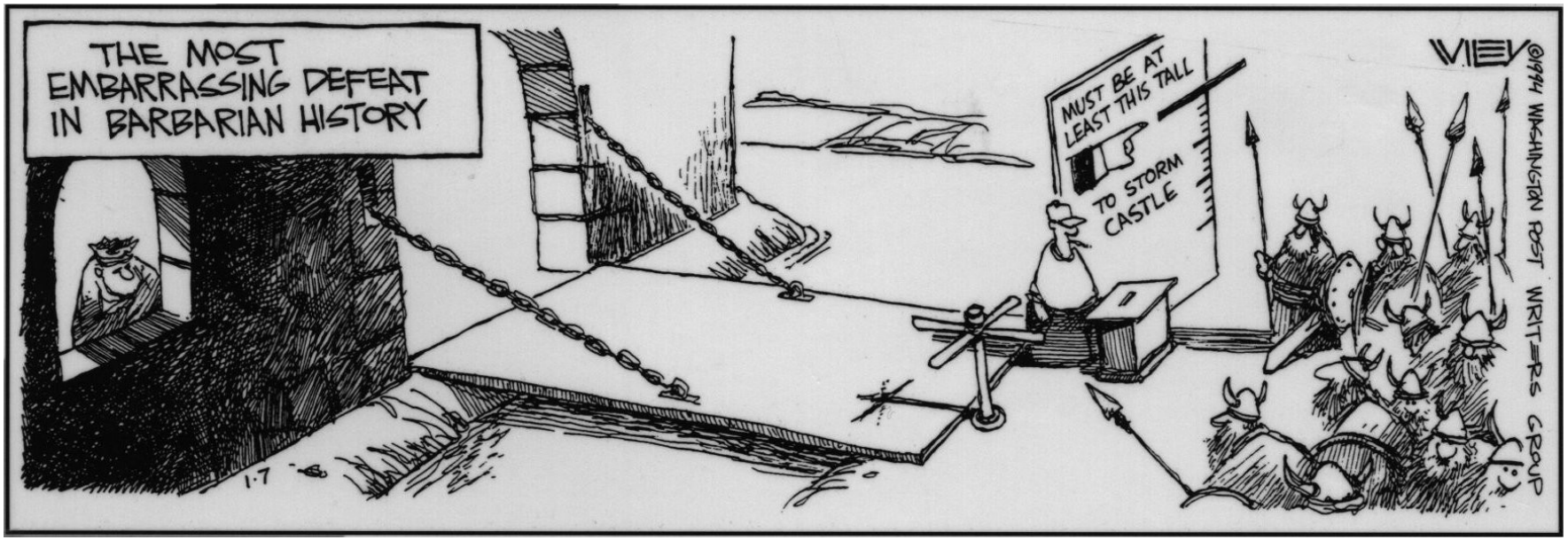
Interesting fields: source and destination port, ACK and SYN bit

TCP Is Connection Oriented

```
10.0.0.251.4354 > 194.239.238.211.echo: S 3633016817
10.0.0.251.4354 < 194.239.238.211.echo: S 2308063735 ack 3633016818
10.0.0.251.4354 > 194.239.238.211.echo: . 1:1(0) ack 1
-- three-way handshake --
10.0.0.251.4354 > 194.239.238.211.echo: P 1:1449(1448) ack 1
10.0.0.251.4354 < 194.239.238.211.echo: . 1:1(0) ack 1449
10.0.0.251.4354 > 194.239.238.211.echo: P 1449:1790(341) ack 1
10.0.0.251.4354 < 194.239.238.211.echo: . 1:1(0) ack 1790
10.0.0.251.4354 < 194.239.238.211.echo: P 1:1449(1448) ack 1790
10.0.0.251.4354 > 194.239.238.211.echo: . 1790:1790(0) ack 1449
10.0.0.251.4354 < 194.239.238.211.echo: P 1449:1790(341) ack 1790
10.0.0.251.4354 > 194.239.238.211.echo: . 1790:1790(0) ack 1790
-- connection teardown --
10.0.0.251.4354 > 194.239.238.211.echo: F 1790:1790(0) ack 1790
10.0.0.251.4354 < 194.239.238.211.echo: . 1790:1790(0) ack 1791
10.0.0.251.4354 < 194.239.238.211.echo: F 1790:1790(0) ack 1791
10.0.0.251.4354 > 194.239.238.211.echo: . 1791:1791(0) ack 1791
```

Now That We Know a Bit About the TCP/IP Stack, How Can We Use This to Protect Our Network?

Packet Constraints



Packet Constraints

For an UDP/TCP packet we have several fields we can filter on. Among others we have:

- source/destination IP-address
- source/destination port number

For TCP we additionally have the SYN/ACK bits

Introducing a Packet Filter

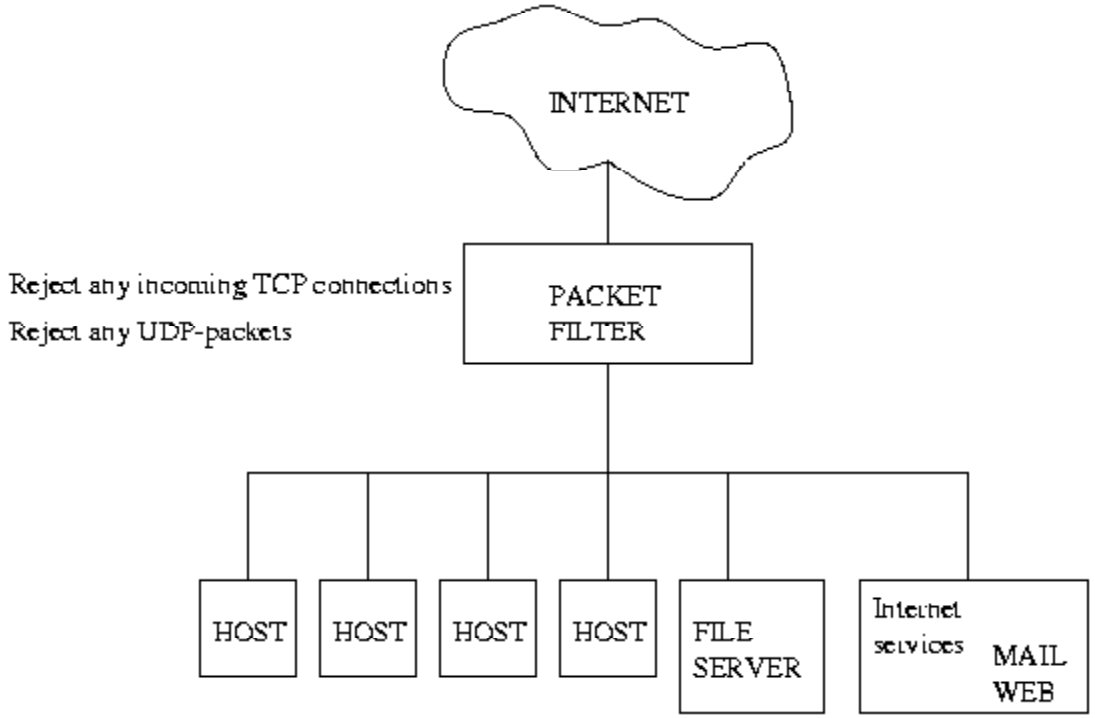
In order to establish a TCP-connection from host A to B, host A must send a SYN flag in the first TCP-packet of the three-way handshake

```
10.0.0.251.4354 > 194.239.238.211.echo: S 3633016817
10.0.0.251.4354 < 194.239.238.211.echo: S 2308063735 ack 3633016818
10.0.0.251.4354 > 194.239.238.211.echo: . 1:1(0) ack 1
```

We can use this in a simple packet filter, which drops any incoming packets with only the SYN flag set. This makes it impossible to make a TCP-connection from the outside.

If the same filter drops all UDP packets as well we have a much more secure system.

A Simple Packet Filter Firewall



This must be really secure...?

What About Our Web Services?

A problem with the previous example is the fact that it is impossible to connect to the web-server, as the web-server is protected as well.

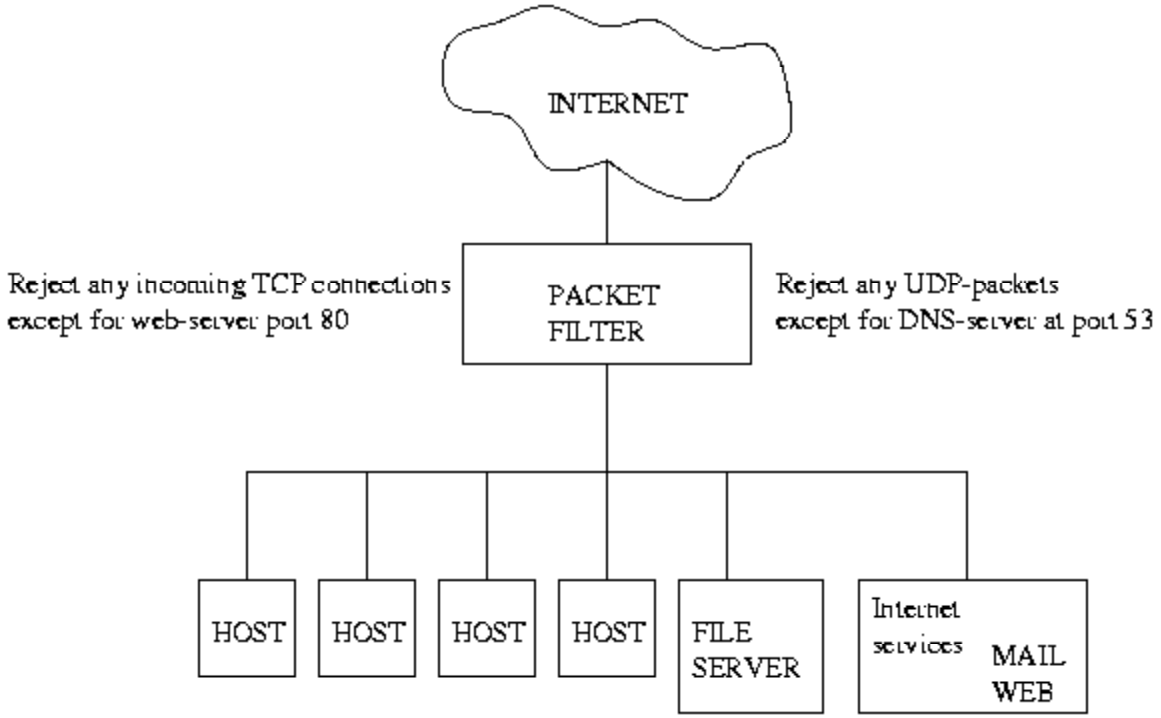
We need an exception to our very restrictive rule set, which allows incoming SYN packets for the webservers IP-adress on port 80 (http)

What About DNS?

Another problem with the previous example is the fact that the protected hosts need access to an external DNS server in order to convert addresses like www.daimi.au.dk to an IP-address like 130.225.16.34.

We need an exception to our rule set, which allows UDP packets to and from at least one external DNS server at port 53 (dns).

New Packet Filter Proposal



This must be really secure...?

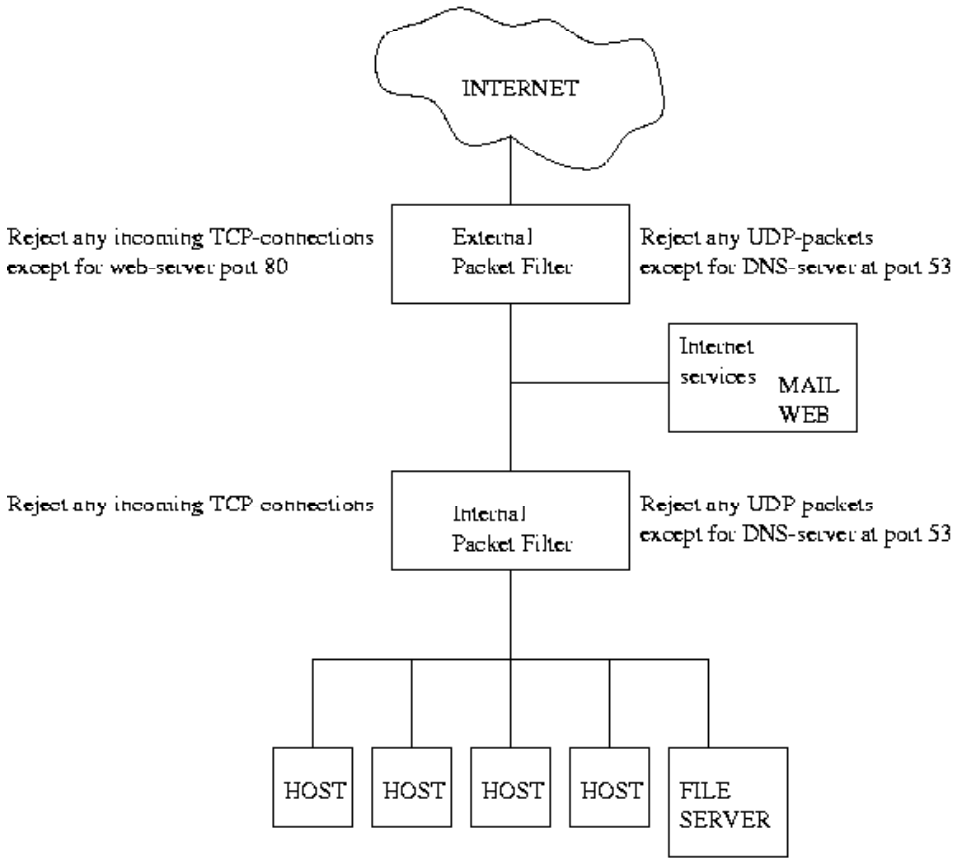
A Single Packet Filter Is Not Enough!

In the previous example, it is impossible to establish a TCP-connection to the protected network, except for connections to the web server at port 80 (http).

But if a hacker finds a hole in the web-server, and gains control of the machine, he can access the other hosts directly!

We must seperate the web-server from the other hosts.

Two Packet Filters Is a Must



Packet Filters Drawbacks

What about all the other nice services offered on the Internet that use the UDP protocol, such as streaming video, computer games etc...

Except for SYN-packets, it is still possible to send arbitrary TCP-packets to the internal hosts.

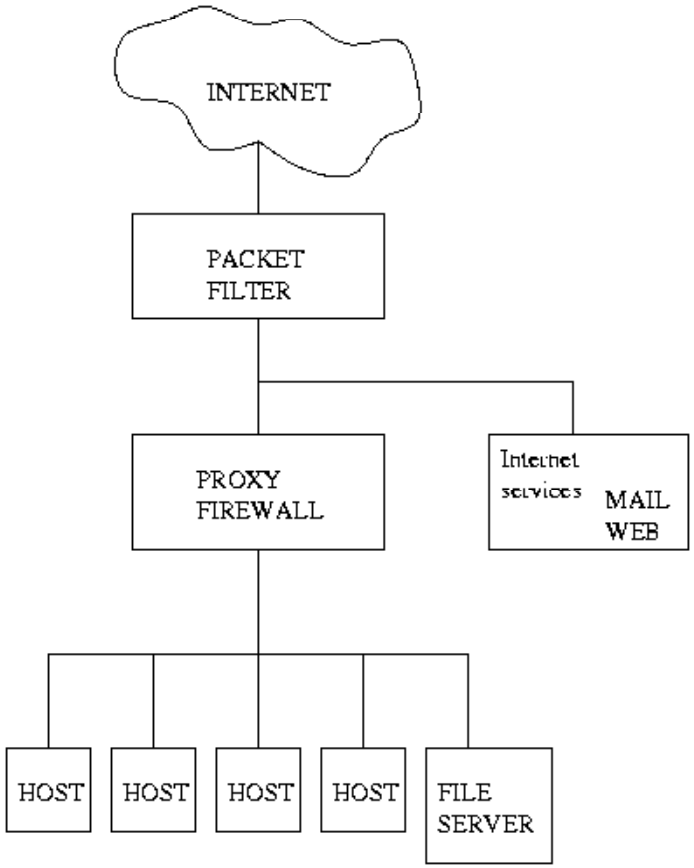
Ideally we want our internal machines to be entirely invisible to the Internet.

Packet filters are not enough, we will have to use something else.

A Proxy Firewall Will Do

The idea behind a proxy firewall is that it acts as a proxy every time an application wish to communicate to the outside world.

Proxy Firewall



Proxy Firewall Advantages

We can safely allow any kind of network traffic from the inside to the outside, as long as we use a proxy to do it.

To the outside it seems that only the firewall exists.

It is impossible to send any network packets directly to the internal hosts or vice versa.

Proxy Firewall Disadvantages

For every network service we wish to use we must install a proxy designed exactly for that service on the firewall.

Furthermore, every network service we wish to use, we must use a client that is able to use a proxy.

What can we do if no proxy exists for a given service?

Proxy Friewall

In general proxy firewalls are considered very secure.

Unfortunately they are not very flexible

Ideally we wish to be able to use any client software.

A Stateful firewall Can Do That

A stateful firewall is an advanced packet filter that keeps track of the state of the network connections going through it.

Whenever a packet arraives to the stateful firewall, it checks whether it matches an ongoing connection. If a match is found the packet can pass through.

NAT and PAT

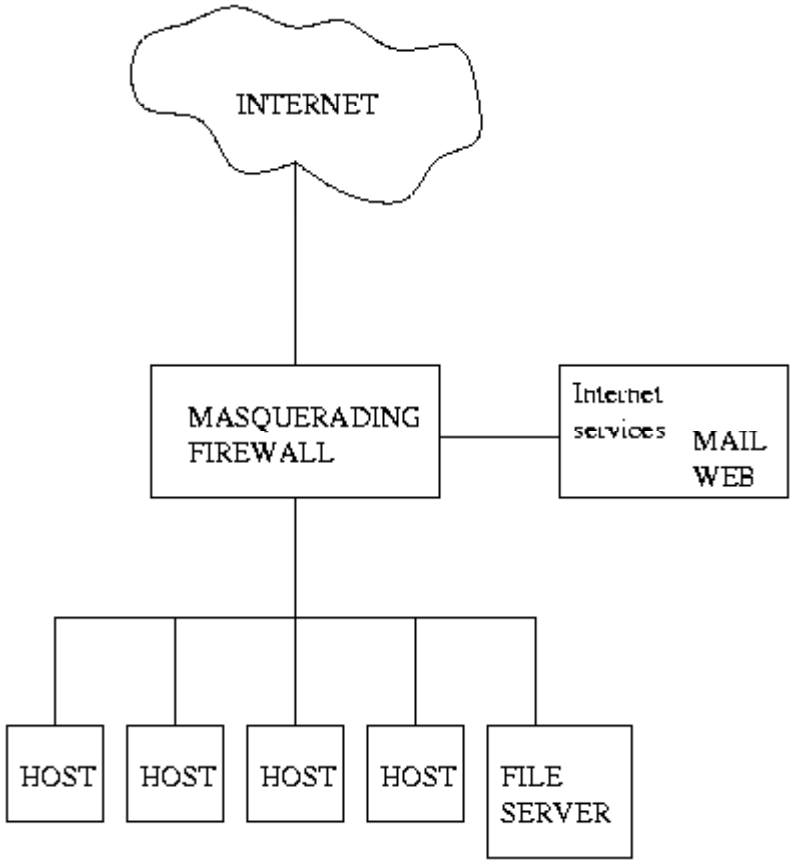
Because the firewall keeps track of all live connections through it, the firewall is able to make both NAT and PAT, or any combination thereof.

NAT: Network Address Translation

PAT: Port Address Translation

A firewall performing NAT or PAT is often referred to as a masquerading firewall.

Masquerading Firewall



Stateful Inspection Takes Us Further

A stateful inspecting firewall is not limited to the network TCP/IP protocols.

For known applications it looks at the application protocol as well.

This enables the firewall to detect when a communication link does something out of the ordinary

It also enables the firewall to filter out certain parts of the data transmitted.

For the HTTP protocol it may filter out javascripts

For the SMTP protocol it may filter out certain types of attachments.

Firewalls come in all shapes and sizes

All the techniques mentioned earlier are used in firewalls today

Firewalls come in many shapes and sizes; the one to use totally depends on the environment in which it operates.

Some firewalls are installed on top of ordinary operating systems, while other firewalls are remotely controlled hardware boxes.

The smallest firewalls protect a single host and might be imbedded in your ISDN router, while the largest firewalls protect thousands of machines.