

## Firewalls

- configuring a sophisticated GNU/Linux firewall involves understanding `iptables`
- `iptables` is a package which interfaces to the Linux kernel and configures various rules for allowing packets and enter and leave the firewall
- `iptables` can be configured as both a
  - packet Filtering Firewall and
  - stateful firewall

## Packet Filtering Firewall

- packet filtering is usually achieved by examining some of the following characteristics of a datagram
- protocol type, for example allow/disallow TCP, UDP, ICMP, DHCP packets
- source address, the machine that supposedly created the packet
  - this can be spoofed

## Packet Filtering Firewall

- destination address
  - the machine the packet to which is being sent
- protocol sub-type
  - some protocol specific sub-type for example TCP's syn packet which is used to initiate a connection
    - or allow ping to operate by allowing the ICMP sub-type (echo-request, echo-reply) packets
- `iptables` also allows you to filter packets based on their state

## Stateful firewall

- provides more control over which packets are allowed through and which are blocked than a stateless firewall
- it is fairly easy to spoof packets and get them to pass through a stateless firewall
  - by contrast it is difficult to do the same across a stateful firewall
- a stateful firewall must keep track of all connections in addition to the normal filtering by port, protocol, and IP address
- in the Linux statefull firewall (`iptables`) a connection can either be in one of the following states: NEW, ESTABLISHED, RELATED, or INVALID

## Superiority of a stateful firewall over a stateless firewall

- if a user wishes to browse the web then the firewall must allow incoming packets from any external <IP address:port 80> to your local <ip address: an unprivileged port>
- these packets can be forged and an attacker can create a false <return address:port 80> and sent it to your <local ip address: unprivileged port>
- problem is that the stateless firewall cannot defend against this attack

## A simple stateless attack

- can take many forms, a simple attack which caused problems in the middle 1990s, is that of IP buffer overflow
- here the attacker takes advantage of the IP datagram structure
  - sends three IP datagram fragments with illegal offsets and length values
    - if the victim does not check the boundaries of the IP fragments then a buffer overflow will occur inside the kernel

## A simple stateless attack

- key datagram field values for fragment 1:
  - offset 0, length 40000
- fragment 2:
  - offset 40000, length 30000
- recall that the maximum legal IP datagram size is 65536 bytes
- result is that the victim will be asked to overwrite 70000-65536 bytes of memory

## The stateful firewall

- the stateful firewall keeps a track of all tcp connections made through the firewall
- the firewall knows which IP addresses are currently being connected to
  - it also knows which unprivileged ports are used either side of the firewall
  - it blocks any packet not using these IP addresses and port numbers and it tracks TCP flags (ack, syn, fin, data etc)
  - removes port and ip address when it sees a TCP.fin occur

## Attack example

- using the same attack example the attacker can send the false fragments, but the stateful firewall will reject the packets unless the attacker correctly guesses the unprivileged port number and destination IP address
- stateful firewalls will forget about established connections if no activity occurs
  - thus an attacker cannot attack a network idle machine
  - an active web browser opens up a different tcp connection for each web page accessed and closes it immediately it is fetched
    - the unprivileged port numbers are randomly chosen

## Ftp problem protocol for stateful firewalls

- FTP is a problem for stateful firewalls because
  - buried inside the FTP protocol the FTP client can request the server to send data to an <IP:PORT> combination
    - IP is normally the client IP, but the port is any unprivileged port
- stateful firewalls are normally set up to block incoming tcp connections
  - FTP and other similar protocols also causes problems for machines performing IP masquerading (or NAT)
- solution is for the firewall to understand more about the application level protocol
  - often called application level firewalls

## Disadvantages of Stateful Firewalls

- they are more complex than stateless firewalls
  - require more memory and to track active connections
  - are harder to administer than a stateless firewall
- some protocols cannot be firewalled by stateful inspection of TCP and IP

## Application-level filters

- modern firewalls use application level filters
  - these proxies can read the data part of each packet in order to make more intelligent decisions about the connection
  - IRC or peer to peer file sharing protocols sometimes try to "hide" on HTTP ports
- while traditional stateful firewalls cannot detect this an application level firewall can detect and selectively block HTTP connections according to content

## Application-level filters

- application level filter based firewalls inspect each packet and decide whether it should be allowed to pass the firewall and continue travelling towards its destination, or be discarded
- some firewalls allow packets to pass according to the context of the connection, and not just the packet header characteristics this deep packet inspection provides a much finer grained control.
- nevertheless this can be defeated by application level protocols which encrypt their data

## Application-level filters

- [examine squid](http://en.wikipedia.org/wiki/Squid_cache) ([http://en.wikipedia.org/wiki/Squid\\_cache](http://en.wikipedia.org/wiki/Squid_cache)) and
- web content filtering [dans guardian](http://packages.debian.org/unstable/web/dansguardian) (<http://packages.debian.org/unstable/web/dansguardian>)
- the Linux kernel can be configured to act as a stateless or stateful firewall through the [iptables](http://www.netfilter.org) (<http://www.netfilter.org>) program