What is a Firewall? and Why we need it.

- A firewall is a “choke point/guard box” of controlling and monitoring the network traffic.
- It allows interconnections between different networks with some level of trust.
- It imposes restrictions on network services (only authorized traffic is allowed).
- It enforces auditing and controlling access (alarms of abnormal behavior can be generated).
- It provides perimeter defence.

Terminology

- DeMilitarized Zone (DMZ): a portion of a network that is not fully protected and separates an internal network from an external network.
- Guard: a host that mediates access to a network to allow or disallow certain types of access on the basis of a predefined policy.
- Filtering firewalls: firewalls that perform access control based on the attributes of packet headers, rather than the content.
- Proxy: an intermediate agent or server that acts on behalf of an endpoint without allowing a direct connection between two end points.
- Proxy Firewall: is a firewall that uses proxies to perform access control. It uses on content and/or header information.
- Network security domain: is a contiguous region of a network that operates under a single security policy.

Security Policies

Before a firewall is designed, implemented and configured, an organisation must define a security policy related to firewalls. E.g:

- Servers in an DMZ are not allowed to make direct connections to an intranet of the private network.
- Network systems of the same organisation across the Internet are not allowed to directly contact any systems in the organisation’s intranet.
- Intranet systems are not allowed to directly contact any systems in the Internet.
- Systems in an DMZ serve as mediators (go-between).
- Do not allow dual interface from DMZ servers directly to intranet systems except inner firewalls.

DMZ and Security Policy

- Complete mediation principle: inner firewalls mediate every access involved with DMZ and intranets.
- Separation of privileges: different DMZ servers running different network functions and firewall machines are different entities than the DMZ servers.
- The outer firewall allows HTTP/HTTPS, anonymous FTP and SMTP access to DMZ server.
Which Internet services does the organisation plan to use or provide?
- Telnet
- WWW
- FTP
- Email
- DNS
- X-windows
- Video Conferencing

Where will the services be used?
- on a local network
- across the Internet
- dial-in from home
- on one subnet
- on different subnets
- totally on a wired environment
- mobile
Study of organisation services
- What additional needs, such as encryption, authentication or dial-in support, may be supported?
- What risks are associated with providing these services and access?
- What is the cost, in terms of controls and impact on network usability, of providing full services with protection?
- Who are going to look after those services?
- Are there any other security servers on your existing network?

Firewall Design Policies
- Generally speaking there are two basic design policies:
  (a) Permit any service unless it is expressly denied
  (b) Deny any service unless it is expressly permitted
- Option (b) is more secure but harder to implement

Firewall Basic Design : Examples
- Address filtering
  - Based on source and/or destination address in IP packets
  - Allowing packets with certain IP addresses to go through
  - Blocking packets with certain IP addresses
- Traffic type filtering
  - Based on the type of traffic in IP header (TCP/UDP port numbers)
  - Allowing certain types to go through. E.g http (port 80)
- Content filtering
  - Based on the content of packets. Blocking packets with some patterns in the content.
- Specific filtering: ICMP message type, TCP SYN and ACK bits

Firewalls: Example

Firewalls: Examples
- \texttt{firewall(p: packet)} \{ /* general form */
  if (p->IP \texttt{allowed-domain}) forward (p);
  else drop (p);
\}
- \texttt{firewall(p: packet)} \{ /* more specific form */
  if (p->IP in allowed-domain) forward (p);
  else drop (p);
\}
- \texttt{firewall(p: packet)} \{ /* combine between filtering and content based */
  if (p->IP NOT in allowed-domain) \& p->content contains \texttt{write})
  drop (p);
  else forward (p);
\}

Step 1: Assume denial of all services except those that are expressly permitted.
Step 2: Answer local security policy questions.
Step 3: Study and understand the firewall design principle (1) and (2)
Step 4: Select a firewall product or build your own firewall that can implement your organisation's firewall policy.
Firewall Types – Packet Filtering Firewalls

- Packet filtering firewalls (PFF) are the simplest form of firewalls.
- A PFFs examine each IP packet (based on information in the packet header) and permit or deny according to predefined rules. They use:
  - Send/Receive Address
  - Protocols
  - Protocol Ports
  - User-defined Bitmasks
- PFFs' possible default policies
  - that not expressly permitted is prohibited
  - that not expressly prohibited is permitted

A Simple PFF: Example

```java
boolean allow(packet) {
  if (!match(packet.source, "130.194.*.*"))
    return false;
  /*
    Only allow packets from 130.194.*.*
  */
  else if (match(packet.source, "140.194.225.*"))
    return false;
  /*
    Allow all packets from 130.194.*.*, except from subnet 225.*/
  else
    return true;
}
```

Example of Packet Filtering Rules

Incoming:
permit 0.0.0.0 130.194.57.*
TCP src >= 1024 dst = 25
permit 0.0.0.0 130.194.56.19
TCP src = 25 dst >= 1024

Outgoing:
permit 130.194.226.* 0.0.0.0
TCP src = 25 dst >= 1024
permit 130.194.227.19 0.0.0.0
TCP src = 80 dst = 25

Stateful Packet Filtering Firewalls

- A stateful packet filtering firewall (SPFF) looks at each packet and applies rules or tests, but the rules or tests applied to each packet may be modified depending on packets that have already been processed or in the case of an application relay it will maintain state by definition.
- A SPFF examines each packet in context
  - keeps tracks of client-server sessions
  - checks each packet if it belongs to one session
- These firewalls are able to detect bogus packets out of context

Firewalls at Transport Layer – Circuit Level Gateway Firewalls

- A circuit level firewall doesn't simply allow or disallow packets but also determines whether the connection between both ends is valid according to the predefined rules
  - Relays two TCP connections without examining contents
  - Enforces security by limiting which such connections are allowed
  - Allowing general outbound connections
  - SOCKS is commonly used for this

Circuit Level Gateway Firewalls

- Whether a connection is valid may be based upon the predefined rules. Eg:
  - Valid destination IP address and/or port
  - Valid source IP address and/or port
  - Allowed time of day (*)
  - Valid protocol (**)
  - Agreed information (***)
Firewalls at Application Level – Proxy Firewalls (PF)

- Use an application specific gateway / proxy to analyse the traffic.
- A proxy server has full access to a protocol => a user requests service from a proxy => the proxy validates request as legal => then actions request and returns result to the user
- The system may need separate proxies for each service => some services naturally support proxy => others are more problematic => custom services generally not supported

Proxy Firewalls (PF) : Example

Firewalls at Different Layers

<table>
<thead>
<tr>
<th>Layer</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>FTP, SMTP, HTTP, RealPlayer, ...</td>
</tr>
<tr>
<td>Presentation</td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td>TCP</td>
</tr>
<tr>
<td>Transport</td>
<td>IP</td>
</tr>
<tr>
<td>Network</td>
<td>Ethernet, FDDI, DLS, Other</td>
</tr>
<tr>
<td>Data Link</td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td></td>
</tr>
</tbody>
</table>

Proxy Firewalls

- PF is useful since it looks through packet contents. E.g., a web application proxy should look at the GET request being sent to your server and ensure that it’s a valid request.
- No traffic travels directly to the server from the client (or vice versa), therefore neither the client nor the server has to worry about the other’s tampering with the structure or options of the network protocol packets.
- PFs may not be able catch bugs in applications.
- PFs may not be able to work with new technologies.

Firewall Functionality

- Packet filtering/Proxy (PFF & PF)
  - Look through source IP address; destination IP address (PFF & PF).
  - Look through TCP/UDP source port/TCP/UDP destination port (PFF & PF).
  - Look through the payload and header and make decision accordingly (PF).
- Analysing log of packets
  - Look at what was dropped to see if unauthorized access is being attempted and take some action.
- Analysing log files to reduce penetration
  - Look at what was forwarded and apply more restrict rules on the new coming packets.

More sophisticated firewalls (MSF)

- It is possible to design, implement and configure more sophisticated type of firewalls.
  - Such MSFs receives protocol units and interprets them. E.g Guard decides what services to perform on users’ behalf, e.g. based on previous transactions, the trust level of outside connections, previous interactions, etc.
  - MSFs can block binary content but allow text, and scan incoming FTP files for viruses, etc.
Firewalls in hardware/software and with other security components

- Firewalls can be implemented in hardware or software or a combination of hardware and software.
- Which function of a firewall should ONLY be implemented in hardware?
- Which function of a firewall should ONLY be implemented in software?
- How do you balance the firewall function between software and hardware?
- Firewalls can be implemented together with other security components?
- Can firewalls be designed to work with other security components?
- Routers.
- VPN.
- NAT
- HMP
- OS kernel.
- IDS.
- Load balancer.
- Malicious code detection and prevention.

Firewall Implementation

- Should a FE be implemented with VPN on one host?
  - Advantages? Disadvantages?
- Should a FE be implemented with NAT on one host?
- Should a FE be implemented with another system?
  - Should a FE be implemented with router together?
- Should firewalls be transparent to the network?
  - Yes but it is hard and expensive.
- Should a firewall engine (FE) be part of the OS kernel?
  - What are the advantages and disadvantages?
    - (e.g. iptable in Redhat).
- Should a FE be implemented as a user process?
  - Advantages? Disadvantages?
  - How do you balance the firewall function between software and hardware?

Firewall Configurations

- One can configure firewalls in a variety of ways, depending on his/her organization’s specific security policies and overall operations. For eg.
  - Highly secure
  - Medium secure
  - Light secure
- It is important to be able to configure and test your firewall properly.
  - Q? How do you know if you have configured your firewall properly? As I tested it after I’ve configured it.
  - Q? How did you test your firewall? As I tested my firewall by using the tools and commands I know?
  - Q? How did you test your tools and commands? A= OOPS, I don’t know?

Firewall Configurations - E.g

- Generally speaking, with a high security firewall most traffic is denied and only allow minimum settings:
  - DNS replies
  - DHCP
- Make sure wanted services still work
  - Active mode FTP (passive mode FTP, used by default in most clients, should still work)
  - Remote X Window System clients
Firewall Configurations - E.g

When configure a firewall with medium security level. The firewall should at least deny the following:
- Ports lower than the standard reserved ports, used by most system services, such as FTP, SSH, telnet, HTTP and NIS.
- The NFS server port — NFS is disabled for both remote servers and local clients.
- The local X Window System display for remote clients.
- The X Font server port (by default, xfs does not listen on the network; it is disabled in the font server).

A firewall with light security should not allow the following:
- Telnet – use SSH instead.
- The local X Window System display for remote X clients - unless there is a strong authentication.
- NFS – unless there is a strong authentication.
- Network admin tools.

Considerations must be taken when design and implement firewalls

All the following should be considered carefully in design and implementation of firewalls:
- Security
- Network performance
- Cost
- Reliability
- Availability
- Manageability