Where do we provide security?

- Firewall
  - Application layer
    - S-MIME, PGP, Kerberos etc.
  - Transport layer
    - SSL, TSL.
  - Internet layer
    - IPSEC
Firewalls

- A firewall is a collection of components placed between two networks (external and internal) that collectively have the following properties:
  - All traffic from outside to inside and vice-versa passes through the firewall.
  - Only authorized traffic, as defined by local security policy, will be allowed to pass.
  - The firewall itself is immune to penetration.
- Note: the above are goals, not guarantees!

Firewalls - Security Perimeter Defense
What can a firewall do?

- Protects network from intrusion.
- Provides focus for security decisions.
- Enforces security policies.
- Logs internet activities in an efficient manner.
- It can limit your exposure to external attacks.
- It helps in the “CYA” factor.

What a firewall cannot do.

- Cannot protect you against malicious insiders.
- Cannot protect against connections that do not pass through it.
- Cannot protect against completely new threats.
- Cannot protect against viruses.
Firewall components

- Bastion host – Dual homed. Connected to external and internal networks. Must be highly secure.
- Screening router – A router with some kind of packet filtering.
- Proxy server – A program that deals with external servers on behalf of internal clients. Also known as Application Level Gateway.
- Virtual Private Network (VPN) – Not really a component but is convenient to place these services in a firewall. More about VPN’s later.

Packet Filtering

- Block or allow packets based on rules.
- Filtering based on packet headers and interface it arrives on.
  - Example – Inbound telnet open not allowed.
- Filtering Strategies
  - That which is not explicitly permitted is prohibited.
  - That which is not explicitly prohibited is permitted.
Advantages and Disadvantages of Packet Filtering

**Advantages**
- One screening router can protect entire network
- Can be extremely efficient if filtering rules are kept simple
- Widely available. Almost any router, even a Linux box.

**Disadvantages**
- Rules can get hard to configure and test
- Can reduce router performance
- Cannot enforce some policies. For example, user level control.

Proxy Servers

- Proxy services sit between user on the inside and server on the outside. Instead of talking directly, user and server talk through proxy.
- Allows more fine grained and sophisticated control than packet filtering. For example, ftp server may not allow files greater than a set size.
Advantages and Disadvantages of Proxy Servers

- **Advantages**
  - Can be good at logging
  - Can provide caching
  - Can do intelligent filtering based on “content”
  - Can perform user level authentication
  - Can protect against weak or faulty IP implementations

- **Disadvantages**
  - Not all services have proxied versions
  - May need different proxy server for each service
  - Require modification of client

Network Address Translation

- Network Address Translation (NAT) allows a network to use one set of addresses internally and a different set when dealing with external networks.
- It helps conceal internal network and force connections to go through choke point.
- Router does the extra work required for address translation.
NAT - Example

Router 24.88.48.47 with NAT that masquerades.
Could be a “dual-homed bastion host”

Internet

FTP Client
130.27.8.35

To 24.88.48.47:23
from 130.27.8.35:x

To 192.168.0.10:80
from 192.168.0.20:80

To 192.168.0.20:80
from 192.168.0.30:80

To 192.168.0.30:80
from 192.168.0.40:80

To 130.27.8.35:80
from 24.88.48.47:80

To 24.88.48.47:80
from 130.27.8.35:80

To 130.27.8.35:80
from 192.168.0.40:80

To 192.168.0.40:80
from 192.168.0.10:80

NAT - Example

Router 24.88.48.47 with NAT that masquerades

Internet

Web Host
130.27.8.35

To 24.88.48.47:x
from 130.27.8.35:80

To 192.168.0.20:80
from 192.168.0.30:80

To 192.168.0.30:80
from 192.168.0.40:80

To 130.27.8.35:80
from 24.88.48.47:x

To 24.88.48.47:x
from 130.27.8.35:80

To 130.27.8.35:80
from 192.168.0.20:80

To 192.168.0.20:80
from 192.168.0.30:80

To 192.168.0.30:80
from 192.168.0.40:80

To 130.27.8.35:80
from 24.88.48.47:x

To 24.88.48.47:x
from 130.27.8.35:80

To 130.27.8.35:80
from 192.168.0.20:80

To 192.168.0.20:80
from 192.168.0.30:80

To 192.168.0.30:80
from 192.168.0.40:80

To 130.27.8.35:80
from 24.88.48.47:x

To 24.88.48.47:x
from 130.27.8.35:80

To 130.27.8.35:80
from 192.168.0.20:80
Router Setup with Network Address Translation (NAT)
Addresses 10.0.0.0 and 192.168.0.0 reserved for private networks.

Advantages and Disadvantages of NAT

**Advantages**
- Helps enforce firewall control over outbound connections
- Can help restrict incoming traffic
- Can help conceal internal network configuration

**Disadvantages**
- Interferes with logging
- Could interfere with packet filtering
- Could interfere with encryption and authentication
- Dynamic allocation could lead to broken connections
Firewall Architectures – Dual homed gateway.

- Routing turned off on Dual-homed Host.
- Useful if no traffic from internal network to external network allowed.
- Users have to log into dual homed host to access internet. Implies user accounts – not a good idea!
- Services can also be provided by proxying them.

Firewall Architectures – Screened Host

- Services only provided through bastion host.
- Primary security provided by packet filtering.
- If attacker breaks into bastion host, has access to internal network.
Firewall Architectures – Screened Subnet.

- Reduces impact of break-in into bastion host.
- No single vulnerable point that compromises internal network.

Perimeter Network Example
Firewall Architectures – Variations.
Firewall Architectures – Variations.
Firewall Architectures – Variations.
Bastion Hosts

- There are two basic principles for designing and building a bastion host:
  - Keep it simple
    - Applications and services have bugs!
  - Be prepared for it to be compromised
    - Most likely machine to be attacked on your network. Will be attacked and will be compromised!!
    - Do your best to ensure that the bastion host won’t get broken into, but keep in mind the question, "What if it does?"
Choosing a machine

- What Operating System?
  - This can become a religious discussion!!
- How Fast a Machine?
  - A slower machine is a less inviting target.
  - If compromised, a slower machine is less useful for attacking internal systems or other sites.
  - A slower machine is less attractive for insiders to compromise
- What Hardware Configuration?
- Physical location – very important!

Services provided by the bastion host

- Services that are secure
- Services that are insecure as normally provided but can be secured
- Services that are insecure as normally provided and can't be secured
- Services that you don't use, or that you don't use in conjunction with the Internet
Don't Allow User Accounts on the Bastion Host

- Vulnerabilities of the accounts themselves
- Vulnerabilities of the services required to support the accounts
- Reduced stability and reliability of the machine
- Inadvertent subversion of the bastion host's security by users
- Increased difficulty in detecting attacks

Building a Bastion Host

- To actually build the bastion host follow these steps:
  - Secure the machine.
  - Disable all non-required services.
  - Install or modify the services you want to provide.
  - Reconfigure the machine from a configuration suitable for development into its final running state.
  - Run a security audit to establish a baseline.
  - Connect the machine to the network it will be used on.
Securing the Machine

- Start with a minimal clean operating system installation
- Fix all known system bugs
- Use a checklist
- Safeguard the system logs

Which services should you leave enabled?

- Certain services are essential to the operation of the machine. On a UNIX system, these include:
  - init, swap, and page
    - The three kernel pseudo-processes used to manage all other processes
  - cron
    - Runs other jobs at fixed times, for housekeeping and so on
  - syslogd
    - Collects and records log messages from the kernel and other daemons
  - inetd
    - Starts network servers (such as telnetd and ftpd) when such services are requested by other machines
Which services should you disable?

- Three simple rules:
  - If you don’t need it, turn it off.
  - If you don’t know what it does, turn it off (you probably didn’t need it anyway).
  - If turning it off causes problems, you now know what it does, and you can either turn it back on again (if it’s really necessary) or figure out how to do without it.

- Some services you should turn off
  - NFS and related services - nfsd, biod, mountd, statd, lockd, automount, keyserv, rquotad
  - Other RPC services – ypserv, ypbind, ypupdated, rexd, walld, tftp, bootd, bootpd
  - Other - rshd, rlogind, rexecd, routed, fingerd, uucpd, rwhod, lpd

Installing and Modifying Services

- Install services that may not be provided by the operating system
  - For example, using the TCP Wrapper package to protect services
    - Install package and set up access control files that define which hosts and networks are allowed to access which services.
    - Reconfigure your inetd to run the main TCP Wrapper program (called tcpd) instead of the “real” server.
    - When a request for a service comes in, inetd starts tcpd, which evaluates the request against the TCP Wrapper configuration files. This program decides whether or not to log the request, and whether or not to carry out the request.
    - If tcpd decides that the request is acceptable, it starts the “real” server to process the request.
TCP Wrapper

/etc/hosts.deny
ALL:ALL

/etc/hosts.allow
in.telnetd: 199.77.146 24.88.154.17
in.ftpd: 199.77.146.19 199.77.146.102

• 199.77.146 means 199.77.146.any
• Above, telnet connection is available to all on the 199.77.146.0 subnet, and a single off-subnet host, 24.88.154.17 FTP service is available to only to two local hosts, .19 and .102.
• The format for each line is “daemon:host-list”

Other Steps

- Reconfiguring for Production
  - Reconfigure and rebuild the kernel.
  - Remove all unnecessary programs.
  - Mount as many filesystems as possible to read-only.
- Running a Security Audit
  - Use auditing package – COP, Tripwire, SATAN etc.
- Connect machine
Operating the Bastion Host

- Learn What the Normal Usage Profile Is
- Consider Writing Software to Automate Monitoring
- Watch Reboots Carefully
- Do Secure Backups

Packet Filtering

- Packet filtering lets you control (allow or disallow) data transfer based on:
  - The address the data is (supposedly) coming from
  - The address the data is going to
  - Session and application protocols being used
Besides source, destination and protocol fields, IP packets can also be filtered based on options field:
- Source routing is a no no, but quite often firewalls drop any packet with any option set!
- Fragmented packets can cause DoS and also fool an IDS. Often dropped too!

TCP segments filtered based on source destination address and port number.
Connection open requests can be filtered based on ACK field.
TCP ACK Bit

- If ACK is not set then segment is necessarily a connection open request!

UDP messages

- Like TCP can be filtered based on source, destination address and port number
- However they do not have an ACK bit!! How do we distinguish between legitimate responses and unsolicited and possibly malicious packets?
  - Drop all UDP!
  - Dynamic filtering
Dynamic Filtering of UDP messages

Other

- Filtering ICMP messages
- Filtering RPC
- Filtering other protocols
- ...

SP = source port
DA = destination address
DP = destination port
SA = source address
### Packet Filtering Rules - Examples

<table>
<thead>
<tr>
<th>Policy</th>
<th>Firewall Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>No outside Web access.</td>
<td>Drop all outgoing packets to any IP, Port 80</td>
</tr>
<tr>
<td>Outside connections to Public Web Server Only.</td>
<td>Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80</td>
</tr>
<tr>
<td>Prevent Web-Radios from eating up the available bandwidth.</td>
<td>Drop all incoming UDP packets - except DNS and Router Broadcasts.</td>
</tr>
<tr>
<td>Prevent your network from being used for a Smurf DoS attack.</td>
<td>Drop all ICMP packets going to a “broadcast” address (130.207.255.255 or 130.207.0.0).</td>
</tr>
<tr>
<td>Prevent your network from being tracerouted or scanned.</td>
<td>Drop all incoming ICMP, UDP, or TCP echo-request packets, drop all packets with TTL &lt; 5.</td>
</tr>
</tbody>
</table>

### Specifying Packet Filtering Rules

- You want to allow all IP traffic between a trusted external host (172.16.51.50) and hosts on your internal network (Class C net 192.168.10.0). On a Cisco router, your rules would look like this:
  - access-list 101 permit ip 172.16.51.50 0.0.0.0 192.168.10.0 0.0.0.255
  - access-list 101 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 interface serial 1
  - access-group 101 in
  - access-list 102 permit ip 192.168.10.0 0.0.0.255 172.16.51.50 0.0.0.0
  - access-list 102 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 interface serial 1
  - access-group 102 out
Preventing Spoofing by Packet Filtering

Packet Filtering Approaches

Filtering by address
- Could be fooled by spoofing.

Filtering by service
- Service could be provided on a different port number.
- Generally you need to do both as illustrated by example that follows.
Packet Filtering Example

- Allow inbound and outbound SMTP (so that you can send and receive electronic mail) and nothing else.

<table>
<thead>
<tr>
<th>Rule/Action</th>
<th>Source Address</th>
<th>Dest. Address</th>
<th>Protocol</th>
<th>Port</th>
<th>Direction</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permit</td>
<td>External</td>
<td>Internal</td>
<td>TCP</td>
<td>25</td>
<td>In</td>
<td>Permit</td>
</tr>
<tr>
<td>Permit</td>
<td>Internal</td>
<td>External</td>
<td>TCP</td>
<td>&gt;1023</td>
<td>Out</td>
<td>Permit</td>
</tr>
<tr>
<td>Permit</td>
<td>Internal</td>
<td>External</td>
<td>TCP</td>
<td>25</td>
<td>Out</td>
<td>Permit</td>
</tr>
<tr>
<td>Permit</td>
<td>External</td>
<td>Internal</td>
<td>TCP</td>
<td>&gt;1023</td>
<td>In</td>
<td>Permit</td>
</tr>
<tr>
<td>Deny</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Either</td>
<td>Deny</td>
</tr>
</tbody>
</table>

Packet Filtering Example (Contd.) – Actions on specific packets

<table>
<thead>
<tr>
<th>Packet</th>
<th>Direction</th>
<th>Source Address</th>
<th>Dest. Address</th>
<th>Protocol</th>
<th>Port</th>
<th>Action (Rule)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In</td>
<td>192.168.3.4</td>
<td>172.16.1.1</td>
<td>TCP</td>
<td>25</td>
<td>Permit (A)</td>
</tr>
<tr>
<td>2</td>
<td>Out</td>
<td>172.16.1.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>1234</td>
<td>Permit (B)</td>
</tr>
<tr>
<td>3</td>
<td>Out</td>
<td>172.16.1.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>25</td>
<td>Permit (C)</td>
</tr>
<tr>
<td>4</td>
<td>In</td>
<td>192.168.3.4</td>
<td>172.16.1.1</td>
<td>TCP</td>
<td>1357</td>
<td>Permit (D)</td>
</tr>
<tr>
<td>5</td>
<td>In</td>
<td>10.1.2.3</td>
<td>172.16.3.4</td>
<td>TCP</td>
<td>6000</td>
<td>Permit (D)</td>
</tr>
<tr>
<td>6</td>
<td>Out</td>
<td>172.16.3.4</td>
<td>10.1.2.3</td>
<td>TCP</td>
<td>5150</td>
<td>Permit (B)</td>
</tr>
</tbody>
</table>

Last 2 packets shows what happens if someone from outside attempts to open a connection from port 5150 on his end to the X11 server on port 6000 on one of your internal systems. It is allowed!!!
Packet Filtering Example (Contd.) - Fixing the rules

Filtering based on both source and destination ports takes care of the earlier problem.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Source Address</th>
<th>Dest. Address</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Dest. Port</th>
<th>Rule/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>In External</td>
<td>Internal</td>
<td>TCP</td>
<td>&gt;1023</td>
<td>25</td>
<td>Permit</td>
</tr>
<tr>
<td>B</td>
<td>Out Internal</td>
<td>External</td>
<td>TCP</td>
<td>25</td>
<td>&gt;1023</td>
<td>Permit</td>
</tr>
<tr>
<td>C</td>
<td>Out Internal</td>
<td>External</td>
<td>TCP</td>
<td>&gt;1023</td>
<td>25</td>
<td>Permit</td>
</tr>
<tr>
<td>D</td>
<td>In External</td>
<td>Internal</td>
<td>TCP</td>
<td>25</td>
<td>&gt;1023</td>
<td>Permit</td>
</tr>
<tr>
<td>E</td>
<td>Either Any</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Deny</td>
</tr>
</tbody>
</table>

Packet actions based on modified rules

<table>
<thead>
<tr>
<th>Direction</th>
<th>Source Address</th>
<th>Dest. Address</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Dest. Port</th>
<th>Action (Rule)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In 192.168.3.4</td>
<td>172.16.1.1</td>
<td>TCP</td>
<td>1234</td>
<td>25</td>
<td>Permit (A)</td>
</tr>
<tr>
<td>2</td>
<td>Out 172.16.1.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>25</td>
<td>1234</td>
<td>Permit (B)</td>
</tr>
<tr>
<td>3</td>
<td>Out 172.16.1.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>1357</td>
<td>25</td>
<td>Permit (C)</td>
</tr>
<tr>
<td>4</td>
<td>In 192.168.3.4</td>
<td>172.16.1.1</td>
<td>TCP</td>
<td>25</td>
<td>1357</td>
<td>Permit (D)</td>
</tr>
<tr>
<td>5</td>
<td>In 10.1.2.3</td>
<td>172.16.3.4</td>
<td>TCP</td>
<td>5150</td>
<td>6000</td>
<td>Deny (E)</td>
</tr>
<tr>
<td>6</td>
<td>Out 172.16.3.4</td>
<td>10.1.2.3</td>
<td>TCP</td>
<td>6000</td>
<td>5150</td>
<td>Deny (E)</td>
</tr>
</tbody>
</table>
Packet Filtering Example (Contd.) - Further problems

- What if the attacker uses port 25 as the client port on his end and then attempts to open a connection to your X11 server?

<table>
<thead>
<tr>
<th>Direction</th>
<th>Source Address</th>
<th>Dest. Address</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Dest. Port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>10.1.2.3</td>
<td>172.16.3.4</td>
<td>TCP</td>
<td>25</td>
<td>6000</td>
<td>Permit (D)</td>
</tr>
<tr>
<td>Out</td>
<td>172.16.3.4</td>
<td>10.1.2.3</td>
<td>TCP</td>
<td>6000</td>
<td>25</td>
<td>Permit (C)</td>
</tr>
</tbody>
</table>

Using ACK bit as additional rule to filter

<table>
<thead>
<tr>
<th>Direction</th>
<th>Source Address</th>
<th>Dest. Address</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Dest. Port</th>
<th>ACK Set</th>
<th>Rule/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>In</td>
<td>External</td>
<td>Internal</td>
<td>TCP</td>
<td>&gt;1023</td>
<td>Any</td>
<td>Permit</td>
</tr>
<tr>
<td>B</td>
<td>Out</td>
<td>Internal</td>
<td>External</td>
<td>TCP</td>
<td>25</td>
<td>&gt;1023</td>
<td>Yes Permit</td>
</tr>
<tr>
<td>C</td>
<td>Out</td>
<td>Internal</td>
<td>External</td>
<td>TCP</td>
<td>&gt;1023</td>
<td>Any</td>
<td>Permit</td>
</tr>
<tr>
<td>D</td>
<td>In</td>
<td>External</td>
<td>Internal</td>
<td>TCP</td>
<td>25</td>
<td>&gt;1023</td>
<td>Yes Permit</td>
</tr>
<tr>
<td>E</td>
<td>Either</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Deny</td>
</tr>
</tbody>
</table>

- Attacker cannot set ACK bit as that would be interpreted as packet of existing connection and dropped.
- Now you can convince yourself that packets 7 and 8 will fail.
Characteristics of a Good Packet Filtering Router

- It Should Have Good Enough Packet Filtering Performance for Your Needs.
- It Can Be a Single-Purpose Router or a General-Purpose Computer.
- It Should Allow Rules Based on Any Header or Metapacket Criteria.
- It Should Apply Rules in the Order Specified.
- It Should Apply Rules Separately to Incoming and Outgoing Packets, on a Per-Interface Basis.
- It Should Have Good Testing and Validation Capabilities

Proxies – Reality and Illusion
How Proxying Works

- Details differ from service to service. Usually requires appropriate proxy server software on the server side. On the client side, need one of the following:
  - Using Proxy-Aware Application Software for Proxying.
  - Using Proxy aware Operating System Software
  - Using Proxy-Aware User Procedures for Proxying
  - Using a Proxy-Aware Router
- Each approach has merits and shortcomings.

Application-Level Versus Circuit-Level Proxies

- An application-level proxy is one that knows about the particular application it is providing proxy services for; it understands and interprets the commands in the application protocol.
- A circuit-level proxy is one that creates a circuit between the client and the server without interpreting the application protocol.
Application Gateway

- Proxies - unique for each application
- Users interact with proxy
- No direct connection
- Higher level of security & control
- Good at logging & authentication
- May or may not be transparent to users
- Performance may be an issue

Circuit Level Proxy (Gateway)

- Client connects to port on proxy (gateway)
- Permission granted by port address
- Creates circuit between client & server without interpreting application protocol
  - Relays connections
  - Relay program copies bytes back & forth
  - Relay services do not examine the bytes
- Can controls connection on basis of source/destination.
- Can also do access control at connect time based on user, department etc. and service being requested.
Using SOCKS for proxying

- SOCKS consists of two parts
  - SOCKS server: Run on a host that can communicate directly with both the Internet and computers inside the firewall.
  - SOCKS client programs: Specially modified Internet client programs that know to contact the SOCKS server instead of sending requests directly to the Internet.

Using the TIS Firewall Toolkit (FWTK) for proxying

- Provides individual proxies for most common internet services.
Microsoft Proxy Server

- Proxying package part of Back Office.
- Provides an HTTP proxy, a SOCKS proxy and a WinSock proxy.

OTHER TYPES OF FIREWALLS

- Stateful inspection
  - Intercepts packets at IP level
  - Provides cumulative data
  - Virtual session info for RPC/UDP applications
  - Proxies are not needed for new applications

- Hybrids
  - Use combination of techniques
  - Rapidly changing
Further Reading

- Firewalls FAQ: http://www.interhack.net/pubs/fwfaq/