Firewalls
Overview

• Background
• General Firewall setup
• Iptables Introduction
• Iptables commands
• “Limit” Function Explanation with icmp and syn floods
• Zone Alarm
What is a Firewall?

- Firewall – a hardware, software, or combination of the two that prevents unauthorized access to or from a private network.
Benefits

• Uninhibited internal LAN traffic

• Ability to leave internal ports open without fear of those ports being abused

• Sense of security by filtering WAN interface for expected traffic
Traffic Control

- Three methods used to control traffic flowing in and out of the network
  - Packet Filtering
  - Proxy Filtering
  - Stateful Inspection
Firewall Configuration

- Rules/filters can be defined to look for a number of things, some of these are:
  - **IP addresses**
  - **Domain names**
  - **Protocols** -
    - IP
    - TCP
    - HTTP
    - FTP
    - UDP
    - ICMP
    - SMTP
    - SNMP
    - Telnet
  - **Ports**
  - **Specific words and phrases**
## What You’re Protected From

<table>
<thead>
<tr>
<th>Security Level</th>
<th>External packets allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>none</td>
</tr>
<tr>
<td>MIDDLE</td>
<td>pre-defined ports (web, ssh) and established connections</td>
</tr>
<tr>
<td>LOW</td>
<td>all packets</td>
</tr>
</tbody>
</table>
What You’re Protected From

• We allow traffic that is expected
  ▪ The firewall is responsible for inspecting connections and packet headers

• We allow all traffic on a few specific ports
  ▪ Certain ports are forwarded to a server
Expected Traffic

- Protects you from floods of packets
  - TCP/SYN, PING/REPLY, IP SPOOFING

- Protects you from scans
  - Port scans and vulnerability probes

- Blocks unwanted connections
  - Telnet, SSH, FTP, and others can be regulated
Port Forwarding

• Biggest security hole in our firewall

• Opened ports to allow traffic to servers
  ▪ All incoming data on this specific port is allowed in, and forwarded to server
    – Hackers could exploit this open port
    – Hackers could exploit a bug in the software on the server
Demilitarized Zone (DMZ)

- Frontline of protection
- “A network added between a protected network and external network in order to provide an additional layer of security”
  -SI Security
- Does not allow external networks to directly reference internal machines
- Acts as system of checks and balances to make sure that if any one area goes bad that it cannot corrupt the whole
Common Firewall Configurations

- Firewall takes care of passing packets that pass its filtering rules between the internal network and the Internet, and vice versa.
- May use IP masquerading but that's all it does.
- Also known as a dual-homed host
- The two "homes" refer to the two networks that the firewall machine is part of
  - one interface connected to the outside home
  - the other connected to the inside home.

http://www.firewall.cx/firewall_topologies.php
Common Firewall Configurations

- The firewall needs only two network cards.
- If you control the router you have access to a second set of packet-filtering capabilities.
- If you don't control the router, your DMZ is totally exposed to the Internet. Hardening a machine enough to live in the DMZ without getting regularly compromised can be tricky.
- The exposed DMZ configuration depends on two things:
  - 1) an external router
  - 2) multiple IP addresses.
- If you connect via PPP (modem dial-up), or you don't control your external router, or you want to masquerade your DMZ, or you have only 1 IP address, you'll need to do something else. There are two straightforward solutions to this, depending on your particular problem.

http://www.firewall.cx/firewall_topologies.php
Common Firewall Configurations

- One solution is to build a second router/firewall.
- Useful if you're connecting via PPP
- Exterior router/firewall (Firewall 1)
  - responsible for creating the PPP connection and controls the access to our DMZ zone
- The other firewall (Firewall 2)
  - is a standard dual-homed host just like the one we spoke about at the beginning
- The other solution is to create a three-legged firewall, which is what we are going to talk about next

http://www.firewall.cx/firewall_topologies.php
Common Firewall Configurations

- Need an additional network adapter in your firewall box for your DMZ.
- Firewall is configured to route packets between the outside world and the DMZ differently than between the outside world and the internal network.
- You can masquerade the machine or machines in the DMZ too, while keeping them functionally separate from protected internal machines.
- The primary disadvantage to the three-legged firewall is the additional complexity. Access to and from the DMZ and to and from the internal network is controlled by one large set of rules. It's pretty easy to get these rules wrong if you're not careful!
- On the other hand, if you don't have any control over the Internet router, you can exert a lot more control over traffic to and from the DMZ this way. It's good to prevent access into the DMZ if you can.
- And I think that just about completes our discussion of Firewall Topologies!

http://www.firewall.cx/firewall_topologies.php
Iptables Introduction

- Iptables is a fourth generation firewall tool for Linux
- Requires kernel 2.3.15 or above with netfilter framework
- Iptables inserts and deletes rules from the kernel’s packet filtering table
- Replacement for ipfwadm and ipchains
How packets traverse the filters

3 default chains: INPUT, FORWARD, OUTPUT
How packets traverse the filters (continued)

• When a packet reaches a circle, that chain determines the fate of the packet
• The chain can say to DROP the packet or ACCEPT it.
• If no rules match in chain, the default policy is used (usually to DROP)
Network Address Translation

The table of NAT rules invoked by ‘iptables –t nat’ contains PREROUTING and POSTROUTING chains

PREROUTING → Routing Decision → POSTROUTING

 Local Process
NAT and iptables

PREROUTING → Routing Decision → FORWARD

INPUT → Local Process → OUTPUT

POSTROUTING
Masquerading

- Special form of Source NAT
- Dynamically changes source address to that of the firewall
- Simple one-line rule

```
iptables -A POSTROUTING -t nat -o eth0 -j MASQUERADE
```
Creating your own rules

- **Adding/Deleting rules:**
  - Append a new rule to an existing chain:
    `iptables -A <chain>`
    ```
    iptables -A PREROUTING -t nat -p tcp -d 1.2.3.4 --dport 80 -j \
    DNAT --to 192.168.1.1:80
    ```
  - Deleting a rule from an existing chain:
    `iptables -D <chain> <rule info>`
    ```
    iptables -D INPUT --dport 80 -j DROP, iptables -D INPUT 1
    ```

- **Changing chains:**
  - Creating a new chain:
    `iptables -N <name>`
    ```
    iptables -N PERMISSION
    ```
Creating your own rules (contd)

- Delete an empty chain:
  ```
  iptables -X <name>
  ```
  ```
  iptables -X PERMISSION
  ```
- List the rules of a chain:
  ```
  iptables -L <name>
  ```
  ```
  iptables -L PERMISSION
  ```
- Flush a chain (delete all rules in a chain):
  ```
  iptables -F <name>
  ```
  ```
  iptables -F PERMISSION
  ```
More iptables commands

- **Specifying jump**
  - If a packet matches a specified rule, jump (-j option) to another chain:
    
    ```
    iptables -A INPUT -j DROP
    ```

- **Specifying protocol**
  - Used to specify the protocol, tcp, udp, or icmp (case sensitive) using -p option.
    
    ```
    iptables -A INPUT -p icmp
    ```

- **Specifying inversion**
  - Used to invert any rules using the ‘!’ option
    
    ```
    iptables -A INPUT -p ! tcp
    ```
Iptables commands (contd)

• Specifying interface
  - Specified with the ‘-i’ (input) or ‘-o’ (output)

    `iptables -A INPUT -i eth0` #check packets coming in on interface eth0

• Specifying source/destination

    `iptables -A INPUT -s 192.168.1.101/24 -d 192.168.1.105`
State matching

- Different states are checked to analyze packets (need to have ip_conntrack module loaded).

- The states that are checked are:
  - NEW: A packet that creates a new connection.
  - ESTABLISHED: A packet belonging to an existing connection (reply or outgoing packet).
  - RELATED: A packet that is related to, but not part of an existing connection (ICMP error).
  - INVALID: A packet that could not be identified.
Port Forwarding

- Using NAT table, destination address is changed based on the port

```
iptables -A PREROUTING -t nat -d 10.1.0.1 -p tcp \
    --dport 80 -j DNAT --to 192.168.1.3:80
```
Defending against ICMP Ping Floods and tcp syn attack

- Using limit module specified with ‘-m limit’ packets can be restricted based on rate of matches

```
iptables -A INPUT -p icmp --icmp-type echo-request \
-m limit --limit 1/s --limit-burst 5 -j ACCEPT
```

Limit burst “recharges” 1 packet every second. This is based on the 1/s limit specified.
Zone Alarm

• Firewall for the Windows OS.

• Several types of alerts:
  ▪ New program alerts: Accept/deny programs to access the internet.
  ▪ Repeat program alerts: grant access permission to program that has already requested before.
  ▪ Server program alerts: grant server permission to a program. Caution: Some Trojan horses require server access to execute.
  ▪ Changed program alerts: If a program has been changed since the last time it access the internet.
What is a zone?

- Zone Alarm classifies computer and networks that you communicate with into good, bad, and unknown zones.

- 3 types:
  - Internet Zone: is the “unknown” zone. All computers and networks belong to this zone until you move them to one of the other zones.
  - Trusted Zone: is the “good” zone. Contains all computers you trust.
  - Blocked Zone: is the “bad” zone. Contains all computers you distrust (only available in Zone Alarm Pro and Zone Alarm Plus version).
What is a zone? (contd.)

• When another computer wants to communicate with your computer – Zone Alarm looks at what zone it belongs to and decides what to do.
Summary

- Firewalls filter unwanted traffic.
- Port Forwarding: big security hole.
- Network Address Translation.
- Use iptables to setup filters.
- State checking.
- Zone Alarm: Firewall for Windows OS.
Acknowledgements


Russell, Rusty, “Linux 2.4 Packet Filtering HOWTO”

Startup script and basis for rules

Steams, William “Adaptive Firewalls with IP Tables”
http://www.ists.dartmouth.edu/IRIA/knowledge_base/adaptive_firewalls.htm

Tyson, Jeff, “How Firewalls Work”
http://computer.howstuffworks.com/firewall.htm/

Young, Scott “Designing a DMZ” http://www.sans.org/rr/firewall/DMZ.php

ZoneAlarm tutorial information provided from
http://www.zonelabs.com