

Firewalls

March 2, 2012

Administrative – submittal instructions

- answer the lab assignment's questions in written report form, as a text, pdf, or Word document file (no obscure formats please)
- email to csci530l@usc.edu
- exact subject title must be "firewallslab"
- deadline is start of your lab session the following week
- reports not accepted (zero for lab) if
 - late
 - you did not attend the lab (except DEN or prior arrangement)
 - email subject title deviates

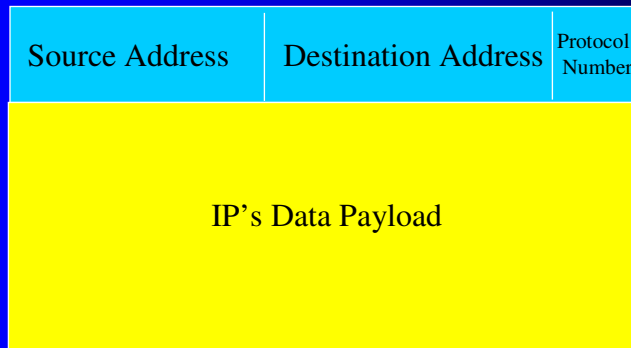
Firewall types

- Packet filter
 - linux, iptables-based
 - Windows XP's built-in
 - router device built-ins
 - single TCP conversation
- Proxy server
 - specialized server program on internal machine
 - client talks to it instead of desired external server
 - it conducts conversation with external server for client and plays relay middleman between them subject to policy
 - 2 separate TCP conversations

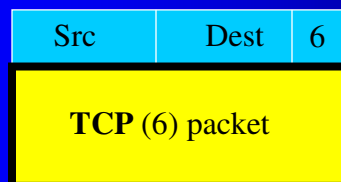
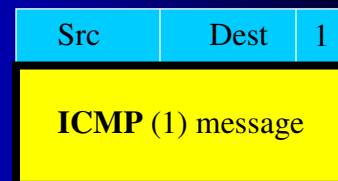
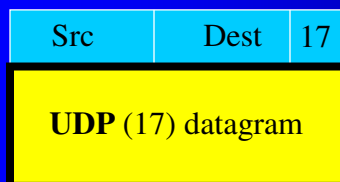
Linux “Netfilter” Firewalling

- Packet filter, not proxy
- Centerpiece command: iptables
- Starting point: packet structure details

IP packet structure



Payload types - subprotocols



... and others

UDP datagram structure

Source Port	Destination Port
UDP's Data Payload	

TCP packet structure

Source Port	Destination Port
Sequence #	Acknowledgment
TCP's Data Payload	

ICMP message structure

ICMP-type	Code	Checksum
header of subject/wayward IP packet or other ICMP-type dependent payload		

Firewall = ruleset

- An in-memory datastructure by whose elements packets that appear at interfaces are evaluated
- A corresponding series of commands, each invocation of which populates the table with a single element
- Elements are called “rules”

Firewall - iptables

- iptables – single invocation creates single rule
- firewall is product of multiple invocations

Iptables organization

- Tables (have chains)
 - filter table
 - nat table
- Chains (contain rules)
 - filter
 - INPUT chain
 - OUTPUT
 - FORWARD
 - nat
 - PREROUTING chain
 - POSTROUTING

An Individual Rule

- condition - examines and qualifies a packet
- action - operates on the packet if it qualifies
- compare – programming language “if” structure

What a Rule says

- “If a packet’s header looks like this, then here’s what to do with the packet”
- “looks like this” e.g.
 - goes to a certain (range of) address(es) or
 - uses the telnet port, 23 or
 - is an ICMP packet
- “what to do” e.g.
 - pass it
 - discard it

```
iptables -t filter -A OUTPUT -o eth1 -p tcp --sport 23 --dport 1024:65535
-s 192.168.4.0/24 -d 0.0.0.0/0 -j ACCEPT
```

– **Table for this rule**

– **Rule action**

- -A add rule to chain/list
- -D delete rule from chain/list
- -P default policy for chain/list

– **Rule chain/list** (tables contain chains)

- INPUT
- OUTPUT
- FORWARD
- PREROUTING
- POSTROUTING

– **Packet qualifiers**

- by interface and direction
- protocol
- source port number(s)
- destination port number(s)
- source address (range)
- destination address (range)

– **Packet disposition**

- ACCEPT
- DROP
- REJECT
- SNAT
- DNAT

What a Chain is

- ordered checklist of regulatory rules
 - Multiple rules, for packets with particular characteristics
 - Single rule for default (catch-all) policy
- operation
 - Packet tested against rules in succession
 - First matching rule determines “what to do” to packet
 - If packet matches no rule
 - Chain’s default policy determines “what to do” to packet

Operationally comparable

```
if [ condition A ]  
    action Alpha; exit  
endif  
  
if [condition B ]  
    action Beta; exit  
endif  
  
if [condition C ]  
    action Gamma; exit  
endif  
.  
.  
.  
action <default>; exit
```

What happens?

← action for first true condition (if any)

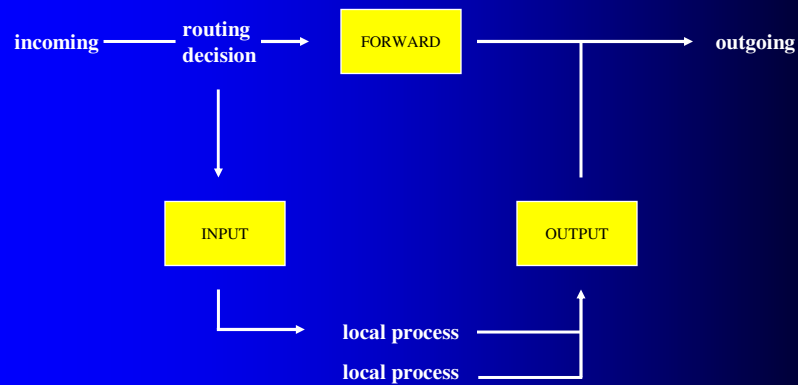
otherwise

default action

Multiple chains

- Input chain
 - When arriving at an interface, do we let a packet come in?
- Output chain
 - When departing from an interface, do we let a packet go out?
- Forwarding chain
 - When traversing this machine to another, do we let a packet pass between interfaces?

Filter traversal by packets



A 4-rule filtering firewall

```
iptables -t filter -A INPUT -i eth1 -p tcp --sport 1024:65535 --dport 23  
-s 0.0.0.0/0 -d 192.168.4.1/32 -j ACCEPT
```

```
iptables -t filter -A OUTPUT -o eth1 -p tcp --sport 23 --dport 1024:65535  
-s 192.168.4.1/32 -d 0.0.0.0/0 -j ACCEPT
```

```
iptables -t filter -P INPUT DROP
```

```
iptables -t filter -P OUTPUT DROP
```

Executed in chronological sequence as shown, resultant 2-chain firewall permits telnet access between this machine 192.168.4.1 and others via eth1. And nothing else.

(0.0.0.0/0 matches any address; aa.bb.cc.dd/32, the single address aa.bb.cc.dd)

Priority of chronology = priority of effect

```
iptables -t filter -A INPUT -i eth1 -p tcp --sport 1024:65535 --dport 23  
-s 64.1.1.1/32 -d 192.168.4.1/32 -j DROP
```

```
iptables -t filter -A INPUT -i eth1 -p tcp --sport 1024:65535 --dport 23  
-s 0.0.0.0/0 -d 192.168.4.1/32 -j ACCEPT
```

```
iptables -t filter -A OUTPUT -o eth1 -p tcp --sport 23 --dport 1024:65535  
-s 192.168.4.1/32 -d 0.0.0.0/0 -j ACCEPT
```

```
iptables -t filter -P INPUT DROP
```

```
iptables -t filter -P OUTPUT DROP
```

... EXCEPT no telnet from machine 64.1.1.1, because first rule eclipses second since it preceded it. (Second not reached, never applied.)

nat table: rules that alter packet

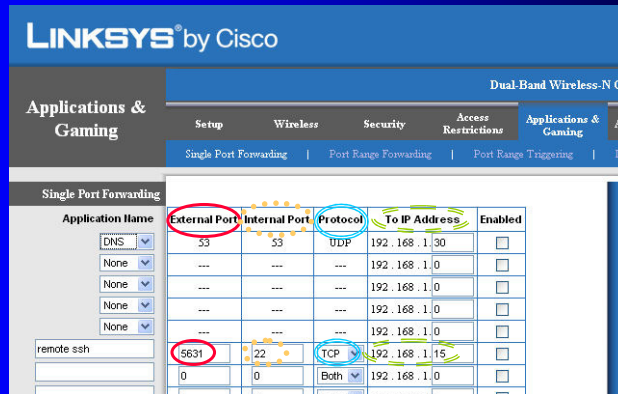
- Masquerading

```
iptables -t nat -A POSTROUTING  
-o eth1 -s 10.0.0.0/8  
-j SNAT --to 216.83.185.193
```

- Pinholing (port forwarding)

```
iptables -t nat -A PREROUTING  
-i eth1 -d 216.83.185.193/32 -p tcp --dport 5631  
-j DNAT --to 10.0.0.15
```

Parallel ways to do the same thing (port forward)



```
iptables -t nat -A PREROUTING  
-i eth1 -d 216.83.185.193/32 -p tcp --dport 5631  
-j DNAT --to 192.168.1.15:22
```

Firewall ruleset philosophies

- **Optimistic/lax** “that which is not expressly prohibited is permitted”
 - set everything open
 - apply selective closures
- **Pessimistic/strict** “that which is not expressly permitted is prohibited”
 - set everything closed
 - apply selective openings

Setting “everything closed” policy

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT DROP
```

```
iptables -P FORWARD DROP
```

Looking further

- conventional filter criteria limited to header fields only
- two further kinds of possible criteria
 - SPI “stateful packet inspection”
 - DPI “deep packet inspection”
- SPI – interrelates packets
 - can tie an incoming packet to an earlier outgoing request, accept for that reason
- DPI – penetrates and examines payload (higher protocol data)
 - can see use of port 80 for non-HTTP traffic, drop for that reason
 - can see use of e.g. peer-to-peer file sharing, drop for that reason
 - tends to overlap with function of intrusion detection software

Firewall persistence

- firewall is memory-resident
- volatile across reboot
- re-erect at boottime by init script containing
 - individual iptables commands or
 - iptables-restore and iptables-save

Start at boot - init script basics

- UNIX has a conventional method to uniformly start/stop services (SysV init)
- one script per service in /etc/rc.d/init.d
- scripts accept parameters start, stop, or restart
- if firewall's script is:
 /etc/rc.d/init.d/firewall
- start it with:
 /etc/rc.d/init.d/firewall start, or
 service firewall start

Avoid vulnerability interval

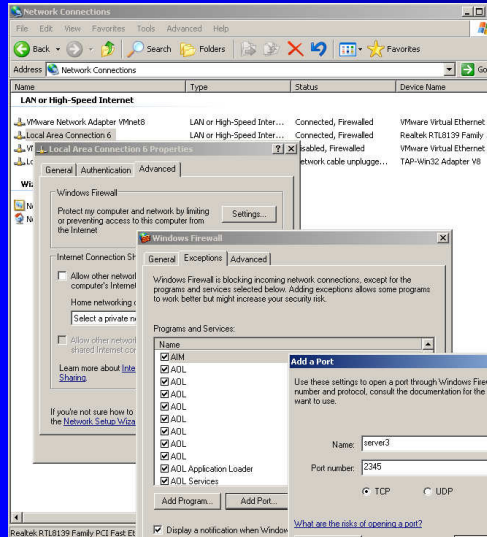
- *first*, call script to erect firewall
 - /etc/rc.d/init.d/firewall
- *only then*, call script to activate/address NICs
 - /etc/rc.d/init.d/network
- calling order controlled by numbering of symbolic links found in /etc/rc.d/rc?.d directories*

*newer "systemd" replacement for SysV init in some linux distributions has a similar After/Before dependency system for ordering startup units

Other packet filter firewalls same

- all are software
- all construct a reference data structure
- all compare packets to structure for decisions
- interfaces differ

Windows XP built-in



an INPUT firewall that's pessimistic with exceptions

equivalent to iptables -P INPUT DROP

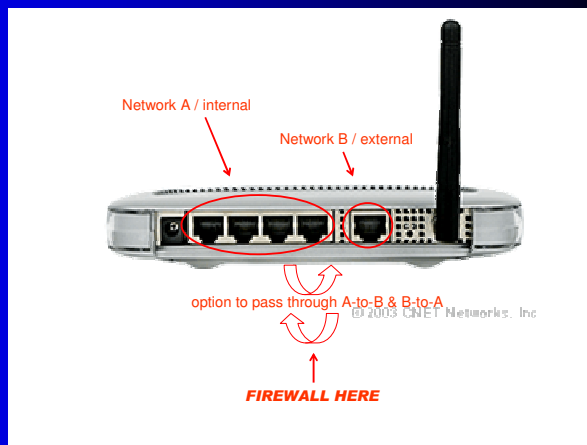
with additional iptables -A INPUT... -j ACCEPT rules for point permission

Netgear WGR614 router built-in



1. Is a computer*
2. Plugs in to *two* LANs

* a router is a computer. It contains a CPU, operating system, memory. It runs software (e.g. firewall!!!) This one has 2 NIC interfaces. Don't be deceived by the lack of keyboard and monitor.





Netgear WGR614 router built-in

an in-to-out FORWARD firewall that's optimistic with exceptions
 equivalent to
 iptables -P FORWARD ACCEPT
 with additional
 iptables -A FORWARD... -j DROP
 rules for point obstruction

Block Services Setup

Service Type:

Protocol:

Starting Port:

Ending Port:

Service Type/User Defined:

Filter Services For :

Only This IP Address:

IP Address Range: to

All IP Addresses

Block Services Setup Help

Services allows you to block Internet access by specific users on your local network based on their IP addresses. In addition, you can prevent the use of certain Internet services completely.

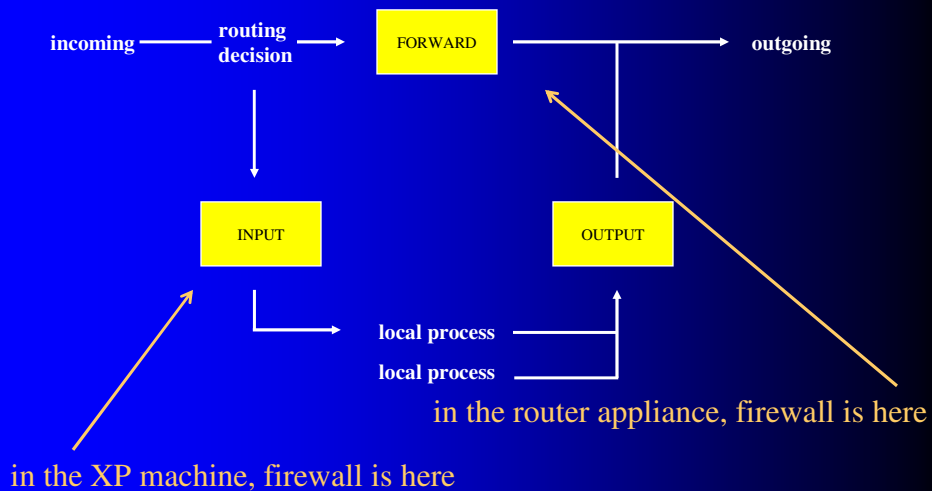
To Add a new Service

- Select the type of service from the pull down list, or select "User Defined" if the desired type is not in the list.
- For "User defined", you must select the protocol, and enter the name and the range of port numbers used by the service. For known services, these fields will be filled in automatically.
- Set the IP address option to determine which PCs are blocked. (See below for more details).
- Click **Apply** to save your changes.

Filter Services For - this determines the computers which will be blocked.

- Only This IP Address - only one (1) PC will be blocked. Enter the IP address of the PC to be blocked.
- IP Address Range - A group of PCs, determined by IP address, will be blocked. Enter the beginning and end of the IP address range of the PCs to be blocked.
- All IP Address - all PCs will be blocked.

Filter traversal by packets



What do these 2 firewalls protect?

- Windows XP
 - the very machine itself that's running XP
- Netgear router
 - not the router itself
 - machines networked to the router
- raises concept of *firewall architecture*
 - what wiring connection “geometry” do you adopt?
 - on which of the computers do you run a firewall?
 - to protect which computers?

Architectures – screened subnet

Building Internet Firewalls, 2nd Edition – Mozilla Firefox

http://safari.adobeexpress.com/1565928717/ch24-16832#X2LudGyYbnF5K1RvYz94bWxpZD0xNTY1OTI4

24.1. Screened Subnet Architecture

The screened subnet architecture, described in Chapter 6, and shown in Figure 24.1, is probably the most common do-it-yourself firewall architecture. This architecture provides good security (including multiple layers of redundancy) at what most sites feel is a reasonable cost.

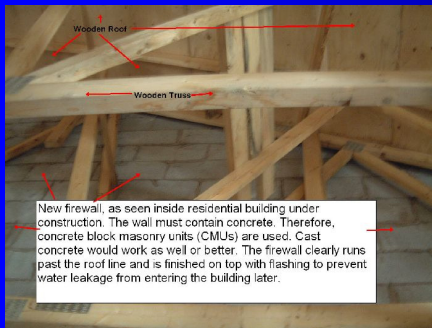
Figure 24.1. Screened subnet architecture

The diagram illustrates the screened subnet architecture. It shows a cloud labeled 'Internet' connected to an 'Exterior Router'. The 'Exterior Router' is connected to a 'Perimeter Network' which contains a 'Bastion Host'. A 'Firewall' is positioned between the 'Perimeter Network' and an 'Interior Router'. The 'Interior Router' is connected to an 'Internal Network' which contains several computer icons.

Why do they call it a hardware firewall?

- it's a firewall
- it's inside a box
- the box is hard

Hardware firewalls



<http://www.pdhone.org/courses/g125/g125.htm>

Ratings of Standard Firewalls, Firewalls, Barriers and Partitions		
Type of Construction	Rating	Configuration
Standard Firewall	4-hour minimum with no openings.	Parapet extends above the roof with wingwalls, end walls or extensions.
Firewall	3 to 4-hour with protected openings.	Parapet extends above the roof with wingwalls, endwalls or extensions.
Fire Barrier	2 to 3-hour with protected openings.	Wall extends from floor to beneath roof or floor deck above.
Fire Partition	1 to 2-hour with protected openings.	Wall extends from floor to ceiling.

But in computer science...

Firewalls are software!

get it?

...it's not hard.

Please see ...

<http://www.iptables.org/>

Linux Firewalls, Michael Rash, No Starch Press, 2007

Linux Firewalls, 2nd edition, Robert Zeigler, New Riders, 2002

Building Internet Firewalls, Zwicky et.al., O'Reilly, 2000