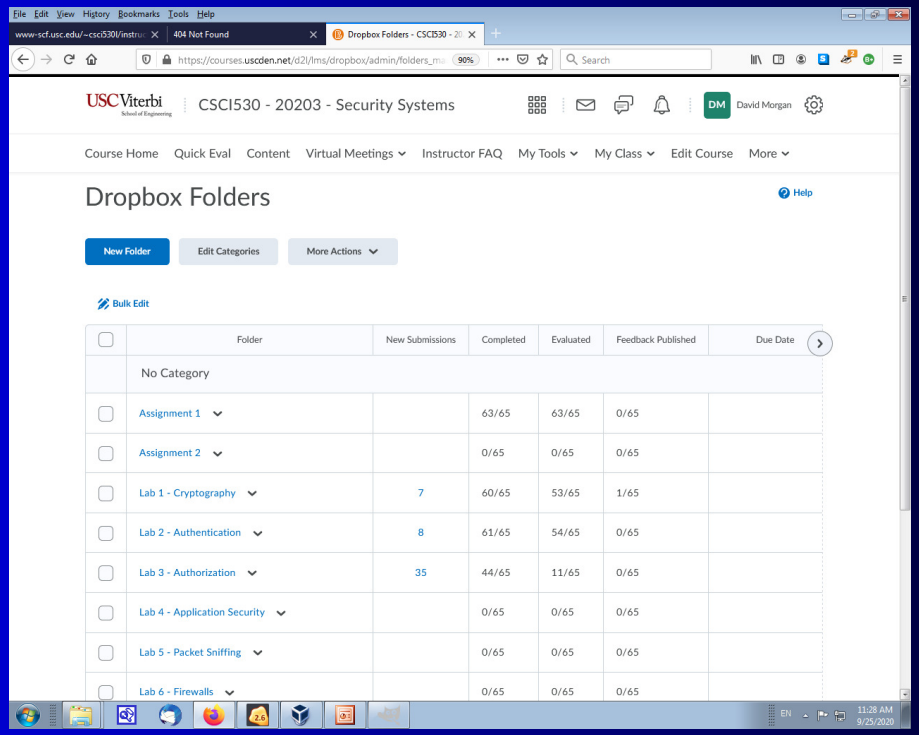
Firewalls

October 16, 2020

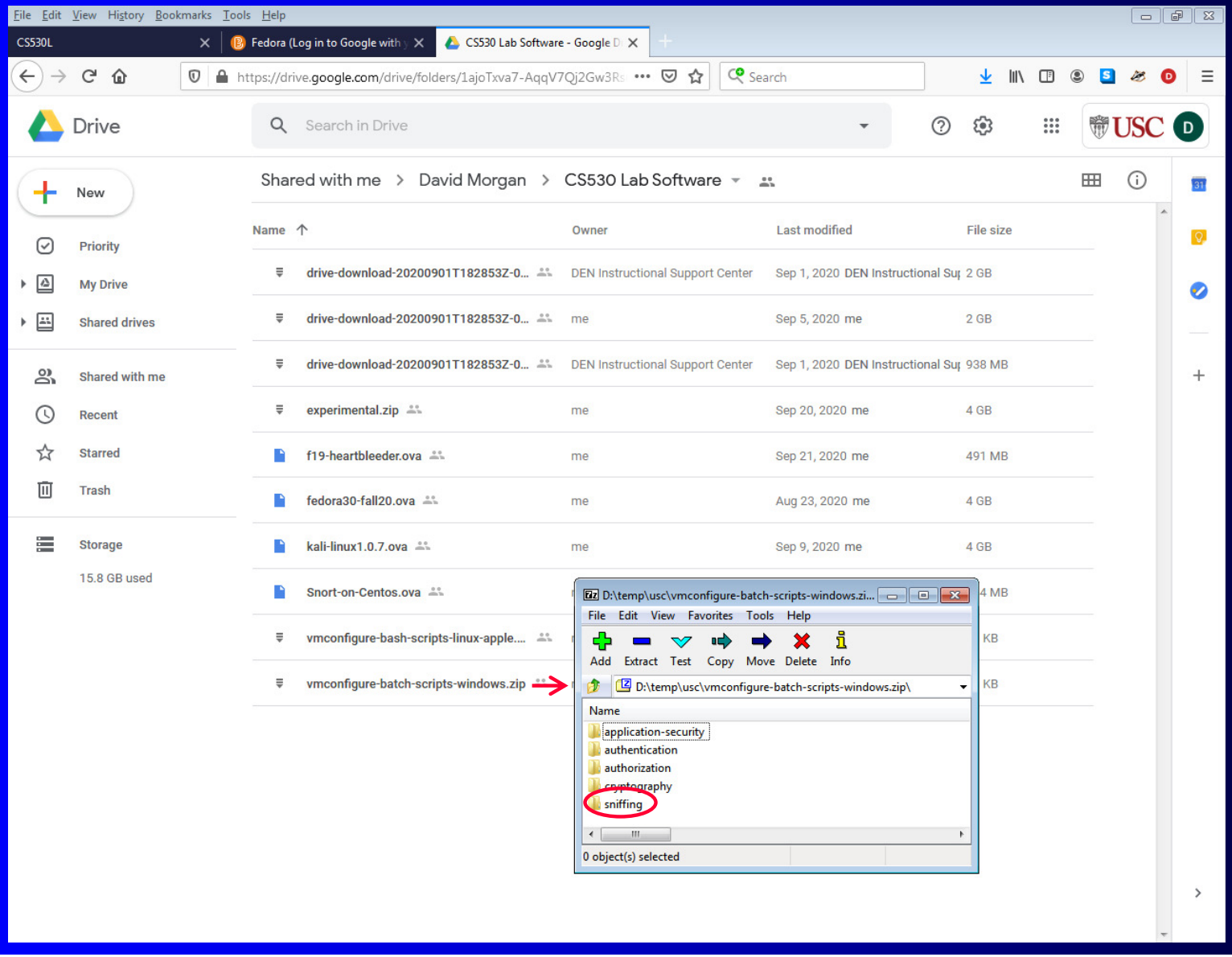
Administrative – submittal instructions

- answer the lab assignment's questions in written report form, as a text, pdf, or Word document file (no obscure formats please)
- deadline is start of your lab session the following week
- reports not accepted (zero for lab) if late
- submit via D2L



Administrative – script files reminder

- re-download the script files' zip
- to obtain the new vmconfigure scripts for this "sniffing" exercise



Firewall types

- Packet filter
  - linux, netfilter-based
  - BSD, PF subsystem
  - Windows's built-in (since XP)
  - 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

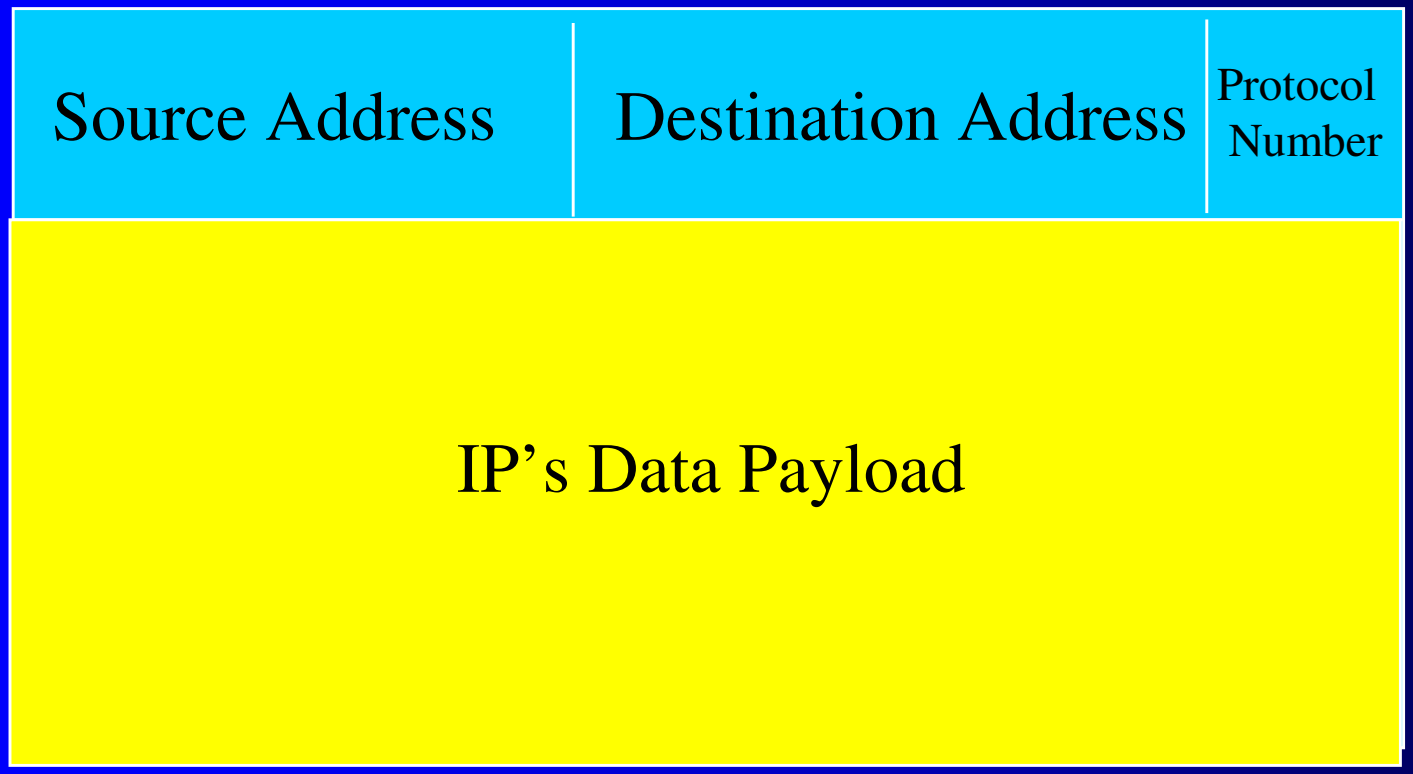
Linux “Netfilter” project

- Netfilter produced iptables, now nftables
- centerpiece commands: iptables, nft
  - nft replaces/extends legacy iptables
  - both coexist in recent linux distributions

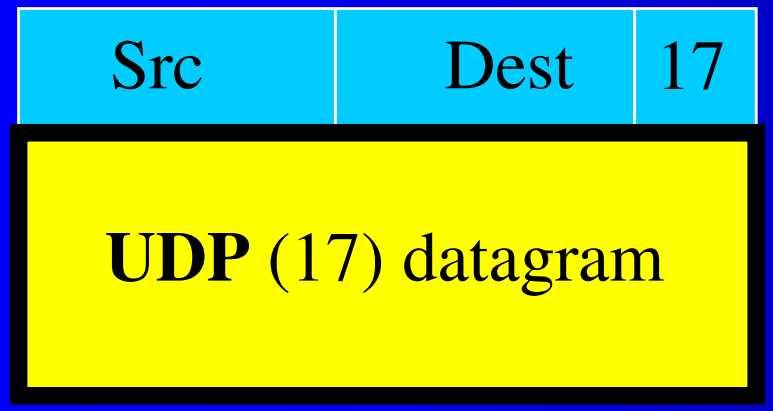
> "nftables replaces the popular {ip,ip6,arp,eb}tables. ... nftables reuses the existing Netfilter subsystems ...there is a backward compatibility layer that allows you run iptables/ip6tables, using the same syntax, over the nftables infrastructure."
> -- https://netfilter.org/projects/nftables/

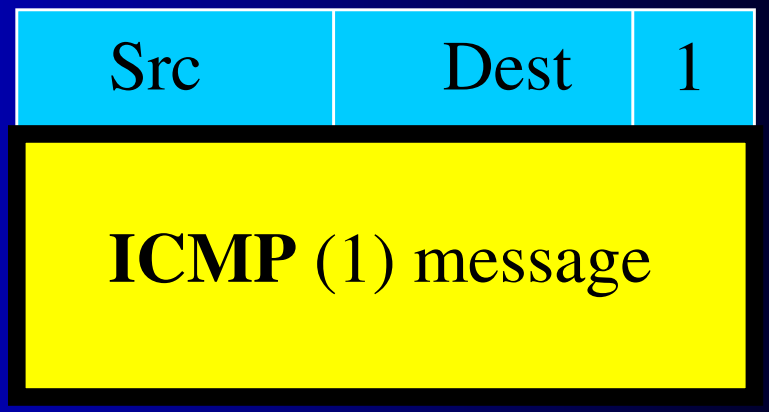
- packet filter, not proxy
- starting point: packet structure details

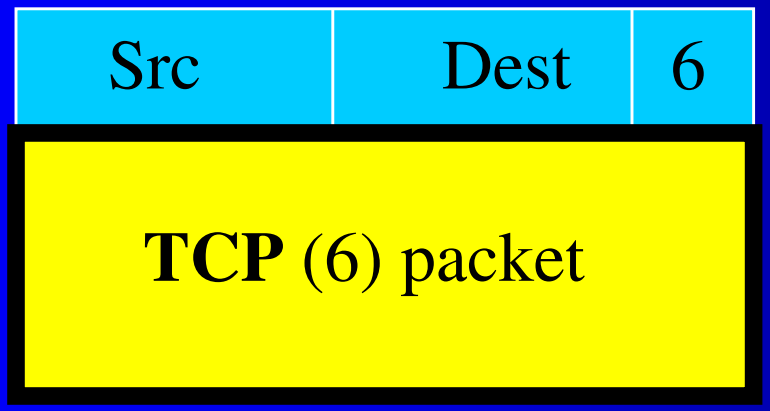
IP packet structure



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 - nftables

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

nftables organization

- tables contain chains
  - chains have types
    - filter type chains
    - nat type chains
  - user creates all chains, none exist by default

- chains contain rules
  - chain types have "hooks"
    - filter type
      - input hook
      - output
      - forward
    - nat type
      - prerouting hook
      - postrouting

sample chain creation syntax: nft 'add chain ip mytable myinputchain { type filter hook input priority 1; policy accept; }'

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

```
nft add rule mytable myoutputchain oifname enp0s3 tcp sport 23 tcp dport
1024-65535 ip saddr 192.168.4.0/24 ip daddr 0.0.0.0/0 accept
```

- action
- object
- target table
- target chain

- 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

What a Chain is

- ordered checklist of regulatory rules
  - multiple rules, for packets with particular characteristics
  - single rule-like 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
```

<u>What happens?</u>

action for first true condition (if any)

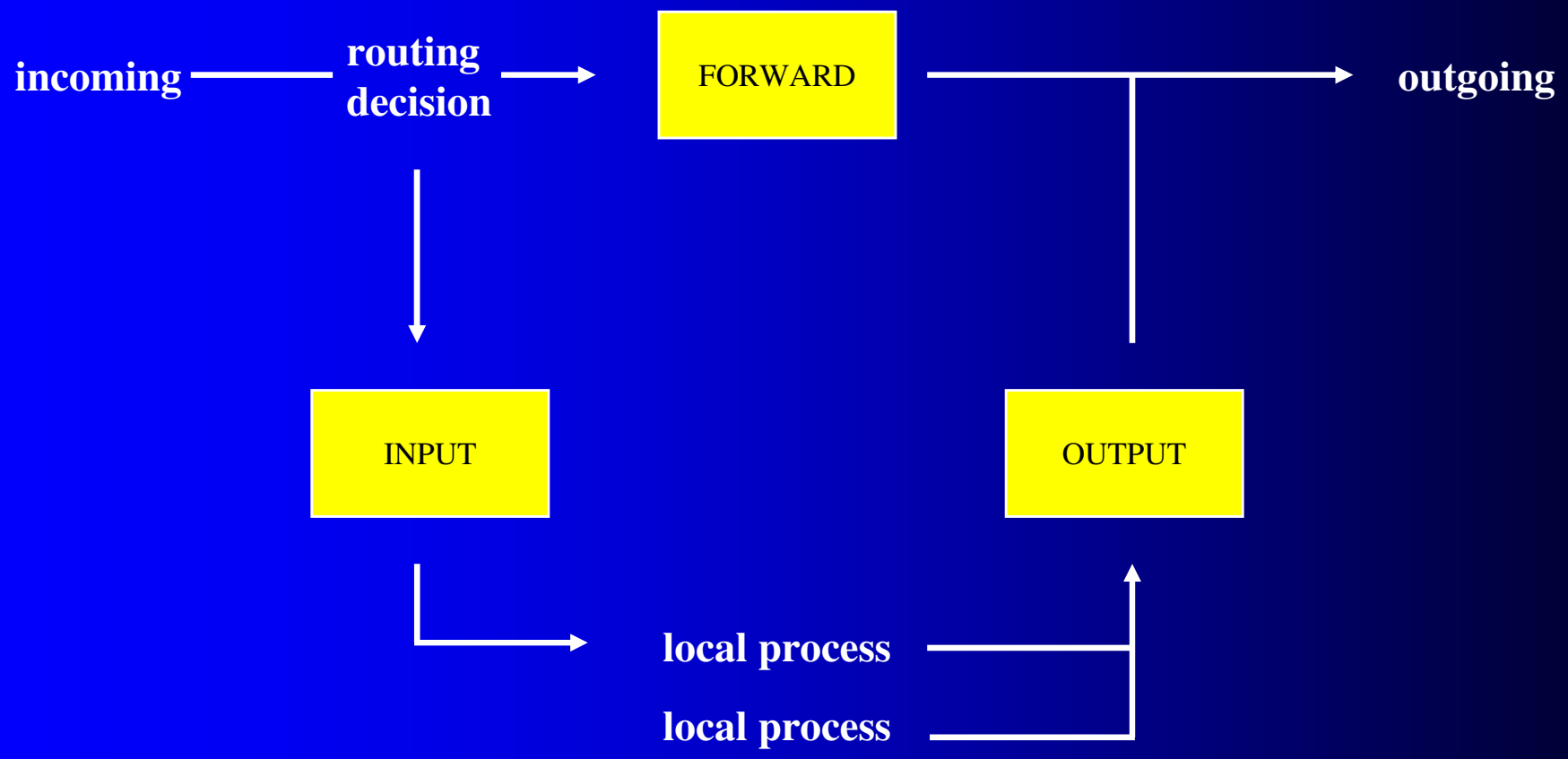
otherwise

default action

Multiple, typical chains

- input-filter chain
  - when arriving at an interface, do we let a packet come in?
- output-filter chain
  - when departing from an interface, do we let a packet go out?
- forwarding-filter chain
  - when traversing this machine to another, do we let a packet pass between interfaces?

Filter traversal by packets



A 2-chain, 2-rule filtering firewall

on telnet server 192.168.4.1

create 2 chains, for input and output, with default "drop"

```
nft 'add chain ip mytable myinputchain   { type filter  hook input    priority 1; policy drop; }'
nft 'add chain ip mytable myoutputchain { type filter  hook output   priority 1; policy drop; }'
```

but accept incoming to port 23 and outgoing from port 23

```
nft add rule mytable myinputchain iifname enp0s3 tcp sport 1024-65535 tcp
dport 23 ip saddr 0.0.0.0/0 ip daddr 192.168.4.1/32 accept

nft add rule mytable myoutputchain oifname enp0s3 tcp sport 23  tcp dport 1024-
65535 ip saddr 192.168.4.1 ip daddr 0.0.0.0/0 accept
```

Executed in chronological sequence as shown, resultant 2-rule firewall permits telnet request into this machine 192.168.4.1 from others via enp0s3, and reply from it out to them. And nothing else.

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

address translations: rules that alter packet

given (table and chains):

```
nft add table mynat
nft 'add chain mynat mypostrouting { type nat hook postrouting priority 100 ; }'
nft 'add chain mynat myprerouting { type nat hook prerouting priority -100; }'
```

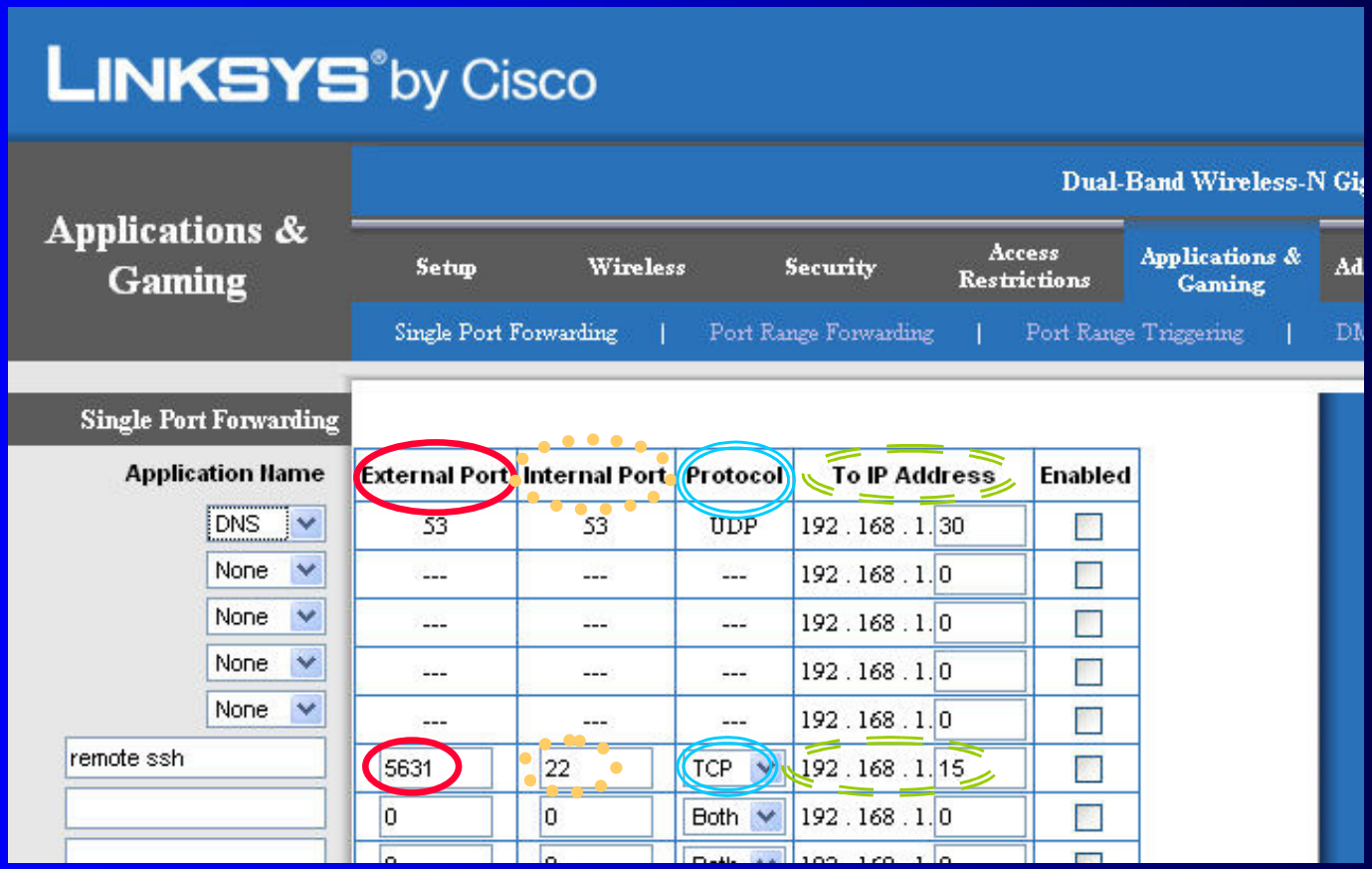
NAT (source network address translation)

```
nft add rule mynat mypostrouting
        ip saddr 192.168.4.0/24 oif enp0s10
                snat 10.0.0.195
```

Port forwarding (destination network address translation)

```
nft add rule mynat myprerouting
        iif enp0s10 tcp dport  23
                dnat 192.168.4.1
```

Parallel ways to do the same thing (port forward)




```
nft add rule mynat myprerouting
        tcp dport 5631 iifname eth1 ip daddr 216.83.185.193
                dnat to 192.168.1.15:22
```

presupposes chain "myprerouting" in table "mynat"

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

root@CLIENT:~

```
[root@CLIENT ~]#
[root@CLIENT ~]# nft flush ruleset
[root@CLIENT ~]# nft add table mytable
[root@CLIENT ~]# nft list table mytable
table ip mytable {
}
[root@CLIENT ~]# nft 'add chain ip mytable myinputchain { type filter  hook input  priority 1; policy drop; }'
[root@CLIENT ~]# nft 'add chain ip mytable myoutputchain { type filter  hook output  priority 1; policy drop; }'
[root@CLIENT ~]# nft 'add chain ip mytable myforwardchain { type filter  hook forward  priority 1; policy drop; }'
[root@CLIENT ~]#
[root@CLIENT ~]# nft list table mytable
table ip mytable {
        chain myinputchain {
                type filter hook input priority 1; policy drop;
        }

        chain myoutputchain {
                type filter hook output priority 1; policy drop;
        }

        chain myforwardchain {
                type filter hook forward priority 1; policy drop;
        }
}
[root@CLIENT ~]#
```

a table with 3 chains
(as yet rule-less)

no frames will pass
(requires alleviating rules for that)

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 prototcol 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 in-kernel memory-resident
- volatile across reboot
- save, then reconstruct at boot time for persistence

```
nft list ruleset > myruleset
nft -f myruleset
```

or

```
nft list ruleset > /etc/sysconfig/nftables.conf
systemctl enable nftables.service
```

s

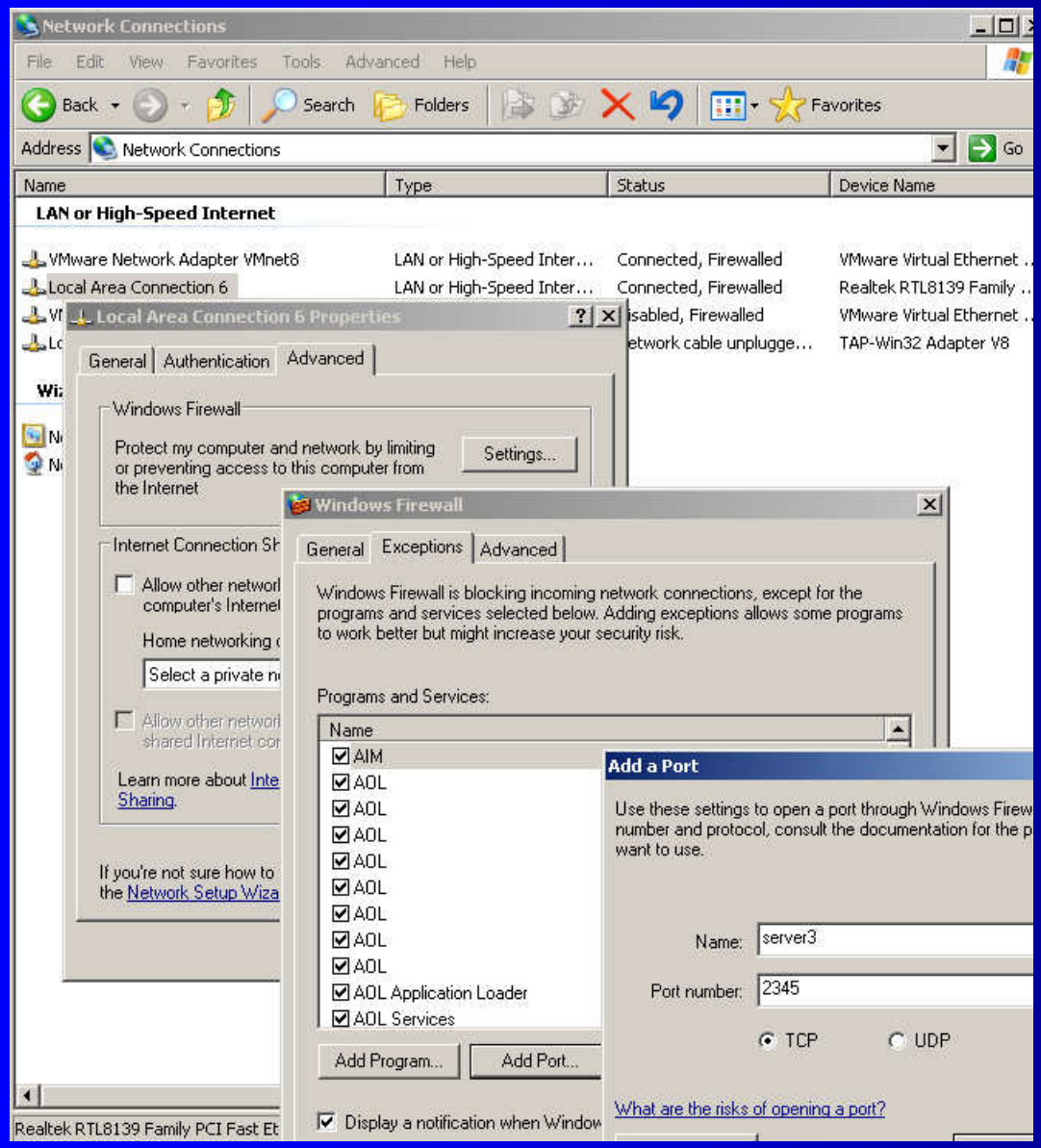
Avoid a vulnerability interval

- *first*, call script to erect firewall
- *only then*, call script to activate/address NICs
- calling order can be controlled through systemd by its 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

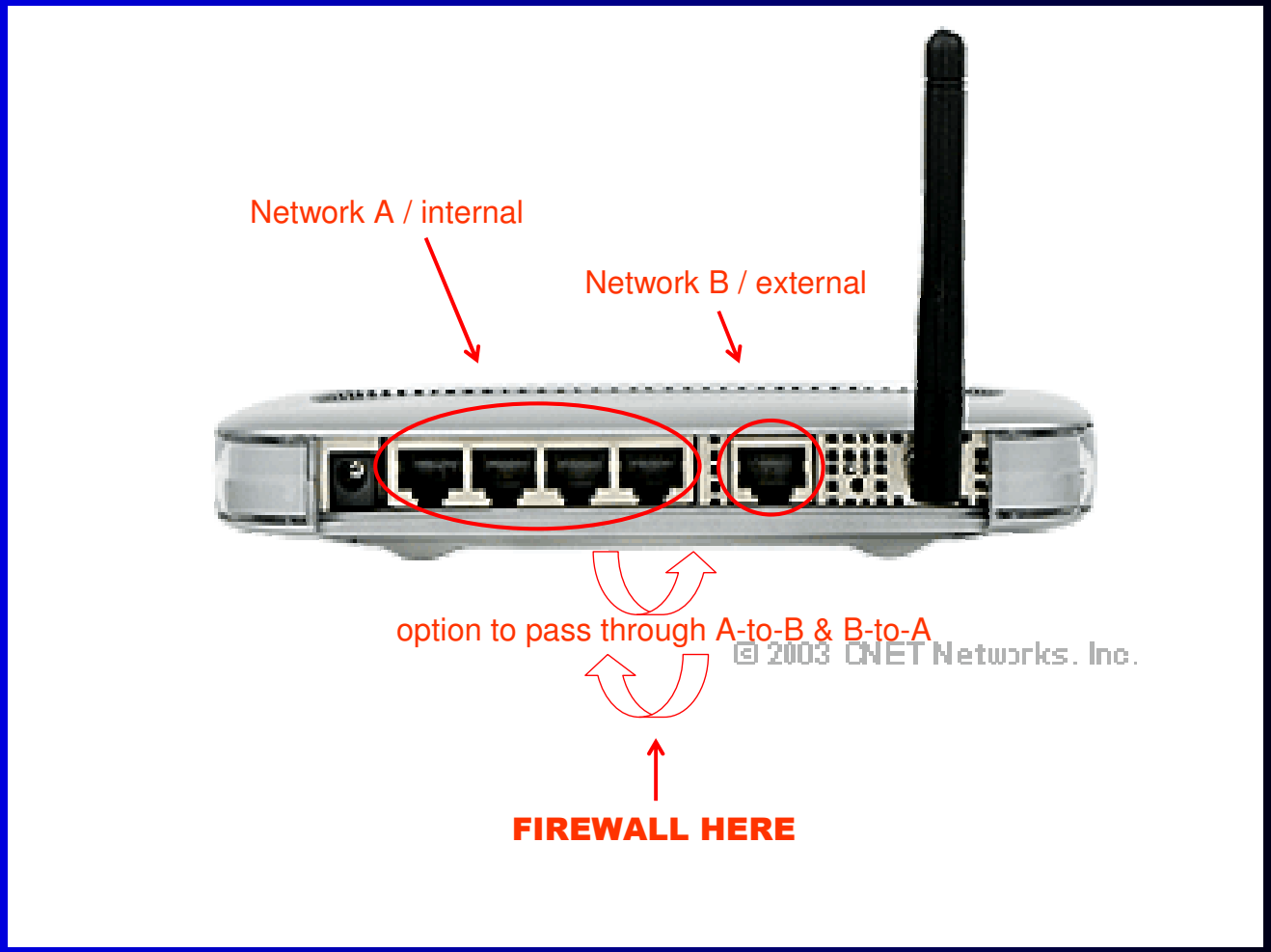


an INPUT firewall that's pessimistic with exceptions

equivalent to "policy drop" in nft chain creation with additional "accept" rules in the chain, for point permission

Netgear WGR614 router built-in

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



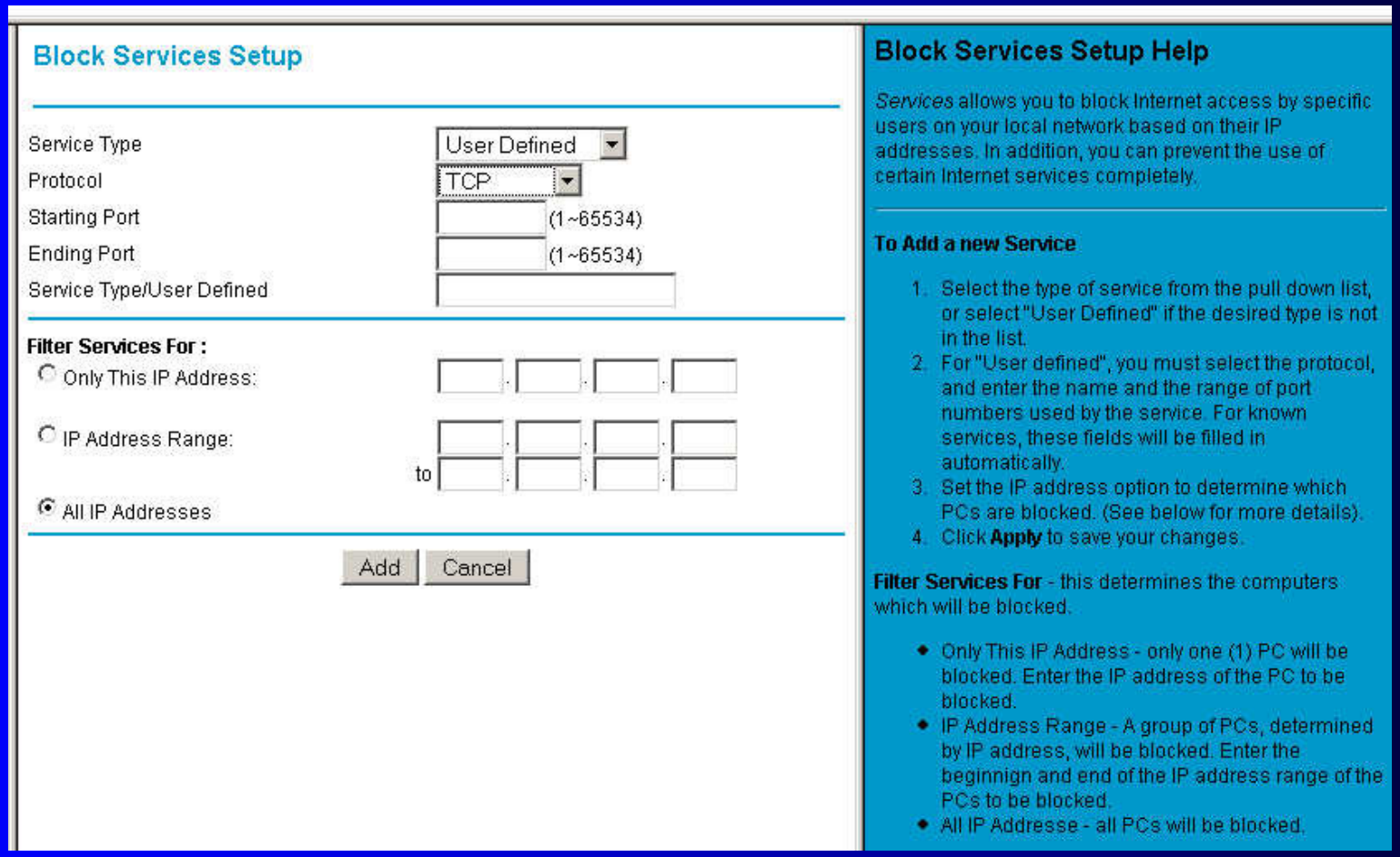
* a router is a computer.
It contains a CPU, operating system, memory. It runs software (e.g. firewall!!) This one has 2 NIC interfaces. Don't be deceived by the lack of keyboard and monitor.

Netgear WGR614 router built-in

an in-to-out FORWARD firewall that's optimistic with exceptions

equivalent to
"policy accept" in chain creation

with additional "drop" rules in the chain,
for point obstruction



Block Services Setup

Service Type	User Defined
Protocol	TCP
Starting Port	(1~65534)
Ending Port	(1~65534)
Service Type/User Defined	

Filter Services For :

- Only This IP Address:
- IP Address Range: to
- All IP Addresses

Add Cancel

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.

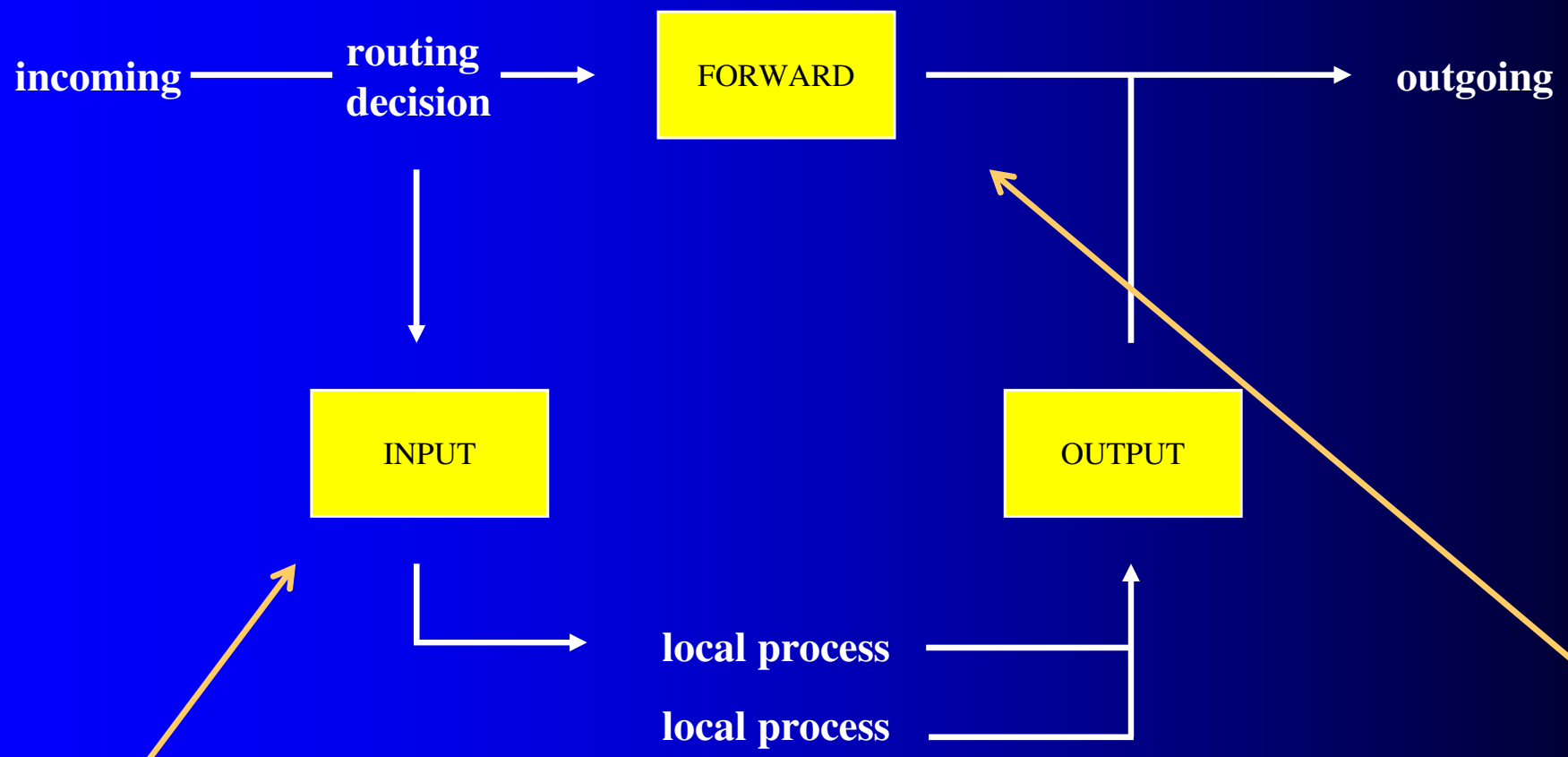
To Add a new Service

1. Select the type of service from the pull down list, or select "User Defined" if the desired type is not in the list.
2. 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.
3. Set the IP address option to determine which PCs are blocked. (See below for more details).
4. 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 beginnign and end of the IP address range of the PCs to be blocked.
- All IP Addresse - all PCs will be blocked.

Filter traversal by packets



in the router appliance, firewall is here

in the Windows machine, firewall is here

What do these 2 firewalls protect?

- Windows
  - the very machine itself that's running Windows
- 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

Bookmarks Tools Help

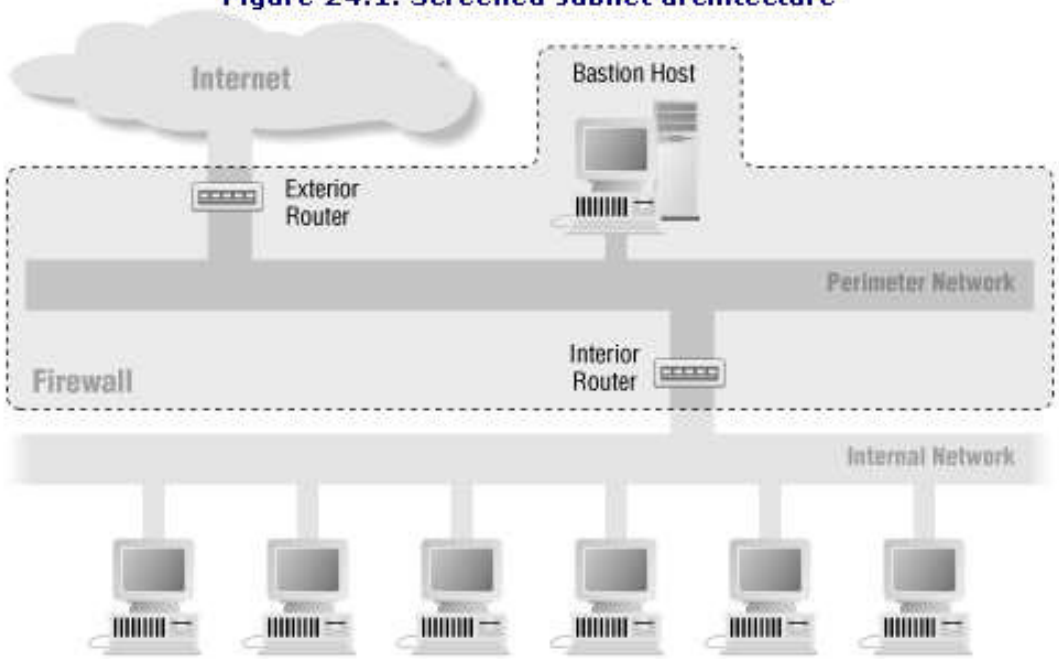
http://safari.adobepress.com/1565928717/ch24-16832#X2ludGVybmFsX1RvYz94bWxpZD0xNTY1OTI4

Forms Images Information Miscellaneous Outline Resize Tools View Source Options

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



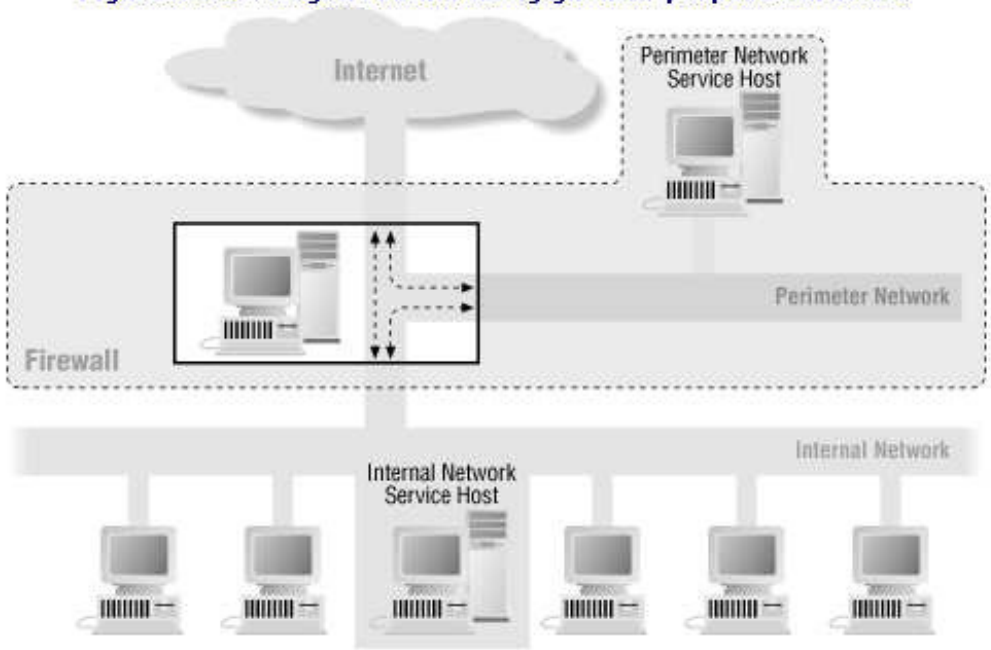
Architectures – merged routers

Building Internet Firewalls, 2nd Edition - Mozilla Firefox

24.2. Merged Routers and Bastion Host Using General-Purpose Hardware

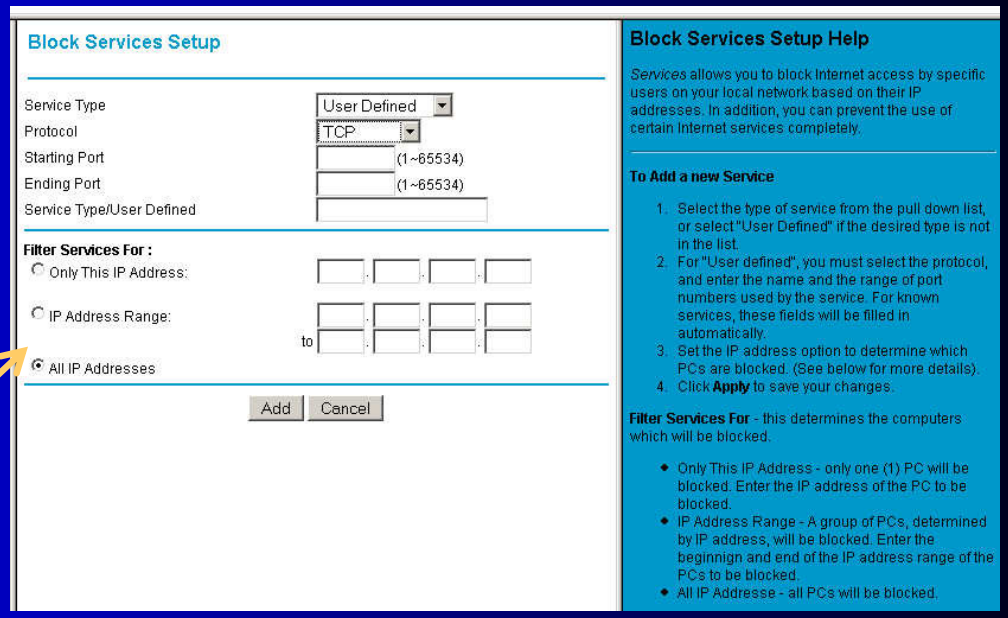
The merged interior and exterior router architecture, described in Chapter 6, and shown in Figure 6.10, is a lower security, lower cost alternative to the screened subnet architecture discussed in the previous section. It can be a very useful architecture for small sites that are facing significant cost constraints, particularly when it is built around a general-purpose computer that can provide not only routing but also flexible packet filtering and proxying. Figure 24.2 shows this architecture.

Figure 24.2. Merged routers using general-purpose hardware



Netgear WGR614 router

the router is not the firewall

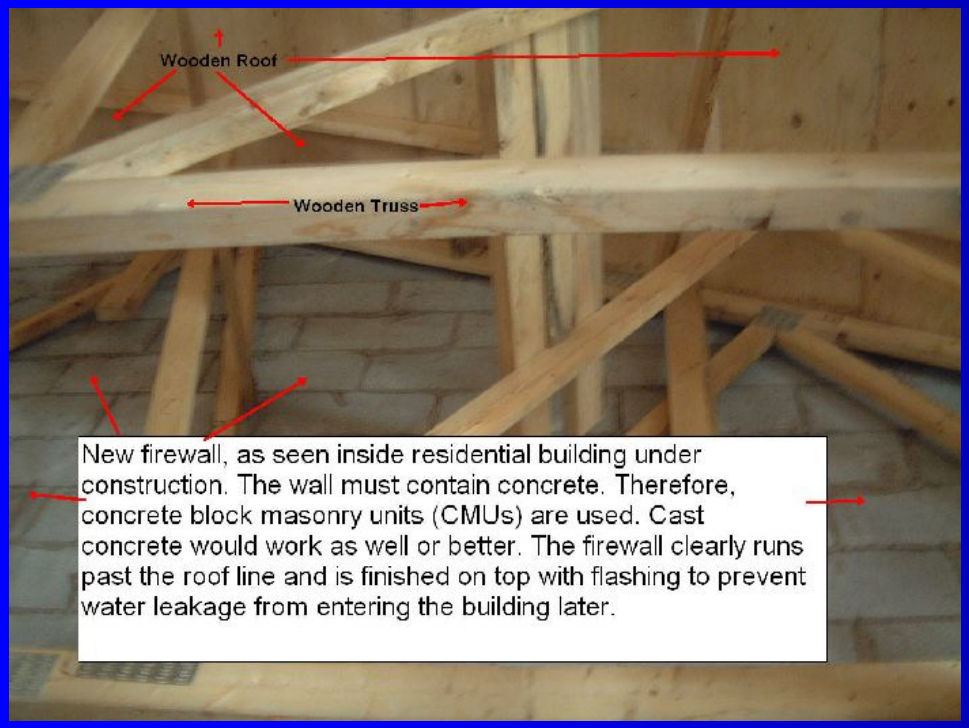


this is (the interface to) the firewall

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.pdhonline.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 so hard.

Please see …

http://www.netfilter.org/

Linux Firewalls, Michael Rash, No Starch Press, 2007

The Book of PF, Peter Nahsteen, No Starch Press, 2008

(PF is an alternative, non-iptables firewall interface tool found in BSD)

Older favorites I learned from, still useful:

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

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