CE 817 - Advanced Network Security

Lecture 3

Mehdi Kharrazi
Department of Computer Engineering
Sharif University of Technology

Acknowledgments: Some of the slides are fully or partially obtained from other sources. Reference is noted on the bottom of each slide, when the content is fully obtained from another source. Otherwise a full list of references is provided on the last slide.

What's a Firewall

- Barrier between us and them.
- Limits communication to the outside world.
  - The outside world can be another part of the same organization.
- Only a very few machines exposed to attack.

- Major firewall vendors: checkpoint, Cisco PIX/ASA

Why Use Firewalls?

- Most hosts have security holes.
  - Proof: Most software is buggy. Therefore, most security software has security bugs.
- Firewalls run much less code, and hence have few bugs (and holes).
- Firewalls can be professionally (and hence better) administered.
- Firewalls run less software, with more logging and monitoring.
- They enforce the partition of a network into separate security domains.
- Without such a partition, a network acts as a giant virtual machine, with an unknown set of privileged and ordinary users.

Traditional Firewalls by Analogy

- Passports are (generally) checked at the border.
- My office doesn’t have a door direct to the outside.
- Any way, what does “firewall” mean? Where does the name come from?

Should We Fix the Network Protocols Instead?

- Network security is not the problem.
- Firewalls are not a solution to network problems. They are a network response to a host security problem.
- More precisely, they are a response to the dismal state of software engineering; taken as a whole, the profession does not know how to produce software that is secure, correct, and easy to administer.
- Consequently, better network protocols will not obviate the need for firewalls. The best cryptography in the world will not guard against buggy code.

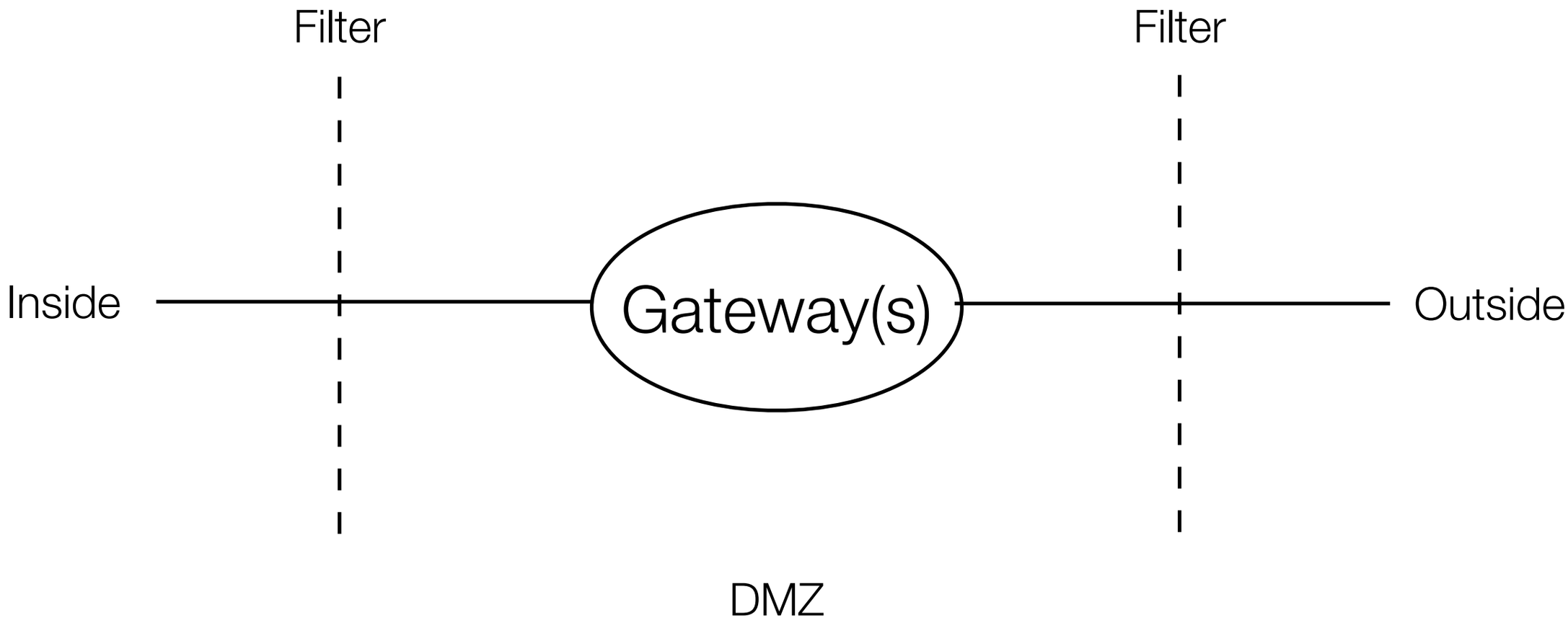
Firewall Advantages

- If you don’t need it, get rid of it.
  - No ordinary users, and hence no passwords for them
  - Run as few servers as possible
  - Install conservative software, don’t get the latest fancy servers, etc.
  - Log everything, and monitor the log files.
    - Keep copious backups, including a “Day 0” backup.
- Ordinary machines cannot be run that way.

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.

Schematic of a Firewall



Conceptual Pieces

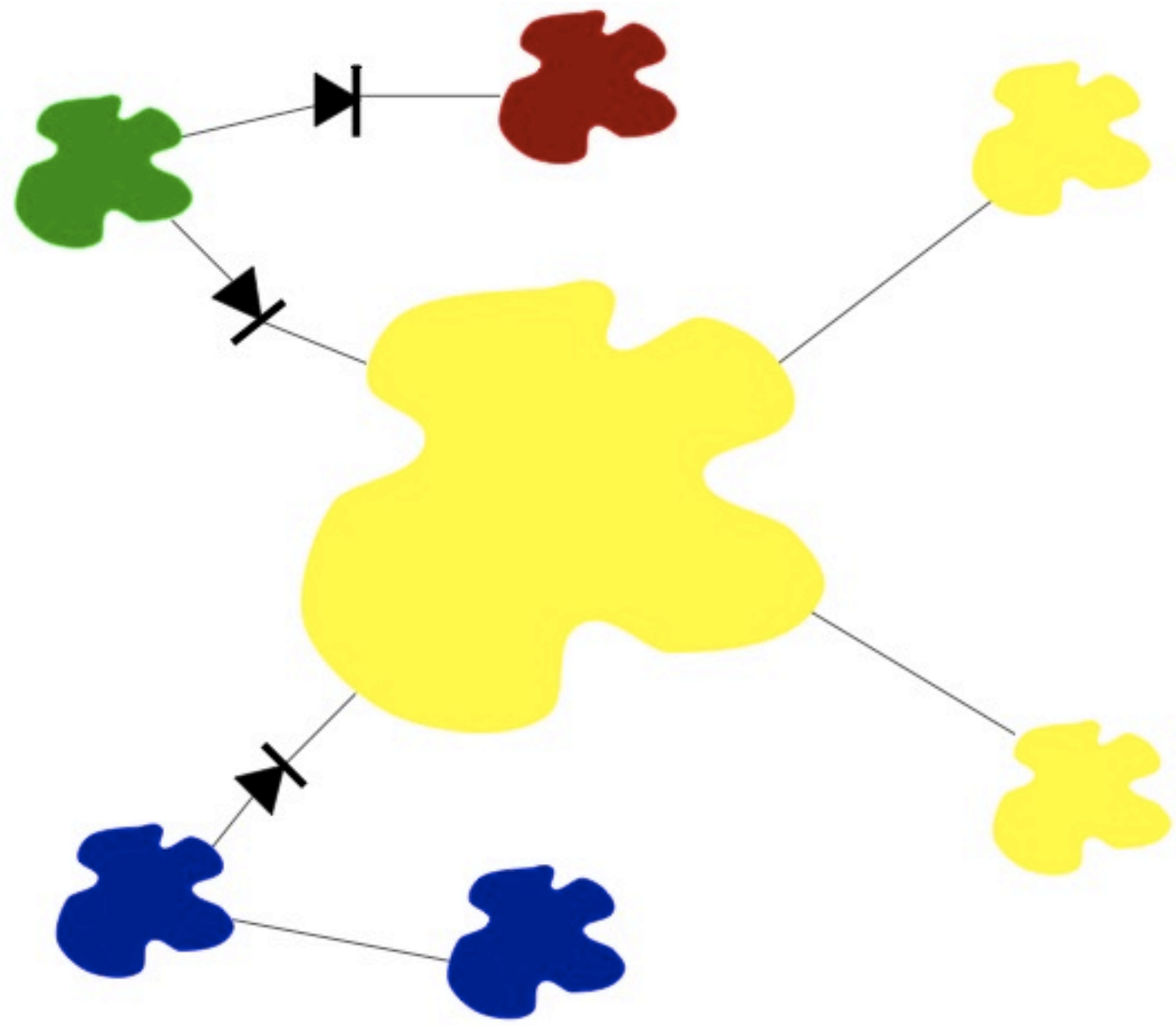
- An “inside” — everyone on the inside is presumed to be a good guy
- An “outside” — bad guys live there
- A “DMZ” (Demilitarized Zone) — put necessary but potentially dangerous servers there

The DMZ

- Good spot for things like mail and web servers
- Outsiders can send email, retrieve web pages
- Insiders can retrieve email, update web pages
- Must monitor such machines very carefully!

Positioning Firewalls

- Firewalls protect administrative divisions.



Why Administrative Domains?

- Firewalls enforce policy
- Policy follows administrative boundaries, not physical ones
- Example: separate protection domains for Legal, HR, Research, etc.

Firewall Philosophies

- Block all dangerous destinations.
- Block everything; unblock things known to be both safe and necessary.

- Option 1 gets you into an arms race with the attackers; you have to know everything that is dangerous, in all parts of your network. Option 2 is much safer.

Blocking Outbound Traffic?

- Many sites permit arbitrary outbound traffic, but. . .

- Internal bad guys?
- Extrusion detection?
- Regulatory requirements?
- Other corporate policy?

Types of Firewalls

- Packet Filters
- Stateful Packet Filters
- Application Gateways
- Circuit Relays
- Personal and/or Distributed Firewalls

- Many firewalls are combinations of these types.

Packet Filters

Packet Filters

- Router-based (and hence cheap).
- Individual packets are accepted or rejected; no context is used.
- Filter rules are hard to set up; the primitives are often inadequate, and different rules can interact.
- Packet filters a poor fit for ftp similar services.

Running Without State

- We want to permit outbound connections
- We have to permit reply packets
- For TCP, this can be done without state
- The very first packet of a TCP connection has just the SYN bit set
- All others have the ACK bit set
- Solution: allow in all packets with ACK turned on

Filtering Rules - Examples

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80
Outside connections to public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a Smurf DoS attack.	Drop all ICMP packets going to a “broadcast” address (eg 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP

Firewall Rules Setup

- Action:
  - Permit (Pass) Allow the packet to proceed
  - Deny (Block) Discard the packet
- Direction:
  - Source (where the packet comes from) <IP Address, Port> or network
  - Destination (where the packet goes) <IP Address, Port> or network
- Protocol:
  - TCP, UDP
- Packet Flags:
  - ACK, SYN, RST, etc.

Sample Rule Set

- We want to block a spammers, but allow anyone else to send email to our gateway.

block: theirhost = spammer

allow: theirhost = any **and**

theirport = any **and**

ourhost = our-gw **and**

ourport = 25.

Incorrect Rule Set

- We want to allow all conversations with remote mail gateways.

Allow: theirhost = any **and**
theirport = 25 **and**
ourhost = any **and**
ourport = any.

Problem?

We don’t control port number selection on the remote host. Any remote process on port 25 can call in.

The Right Choice

Allow: theirhost = any **and**

theirport = 25 **and**

ourhost = any **and**

ourport = any **and**

bitset(ACK).

Permit outgoing calls.

Packet Filters and UDP

- UDP has no notion of a connection. It is therefore impossible to distinguish a reply to a query—which should be permitted—from an intrusive packet.
- Address-spoofing is easy — no connections
- At best, one can try to block known-dangerous ports. But that's a risky game.
- The safe solution is to permit UDP packets through to known-safe servers only.

UDP Example: DNS

- Accepts queries on port 53
- Block if handling internal queries only; allow if permitting external queries

ICMP Problems

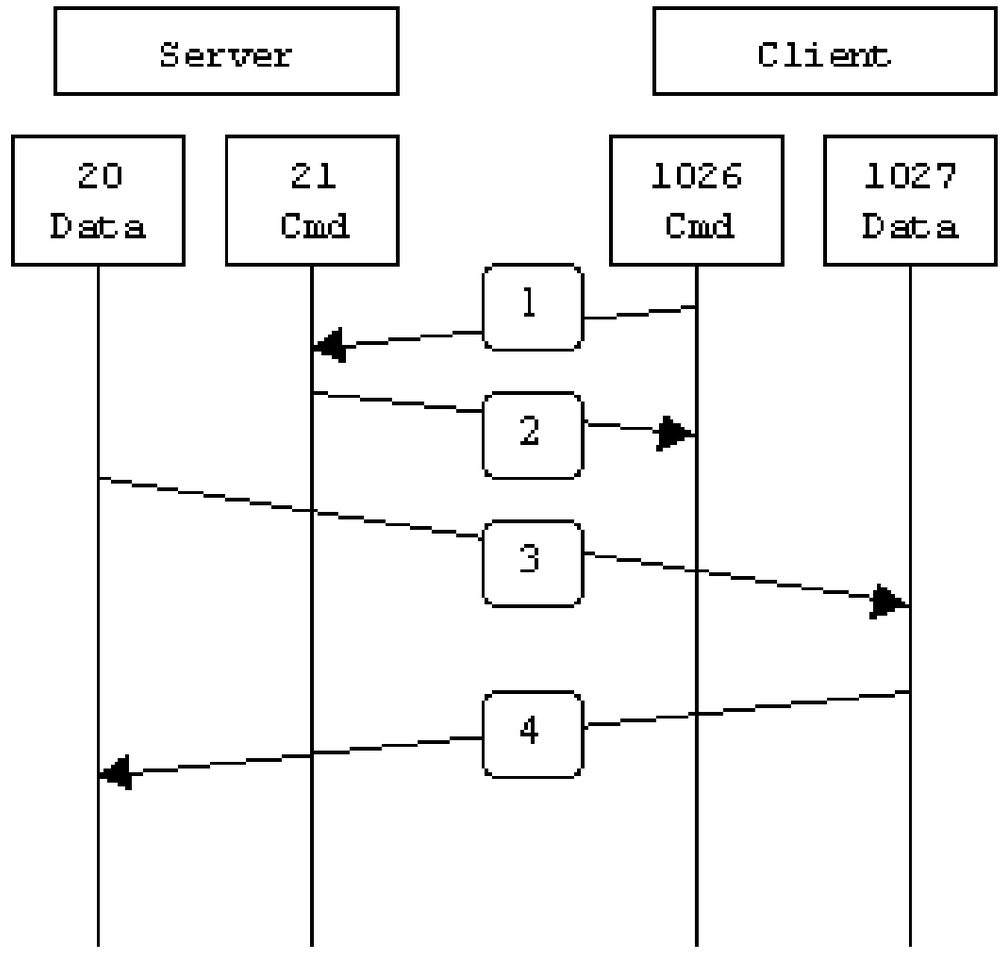
- Often see ICMP packets in response to TCP or UDP packets
- Important example: “Path MTU” response
- Must be allowed in or connectivity can break
- Simple packet filters can’t match things up

FTP, SIP, et al.

- FTP clients (and some other services) use secondary channels
- Again, these live on random port numbers
- Simple packet filters cannot handle this

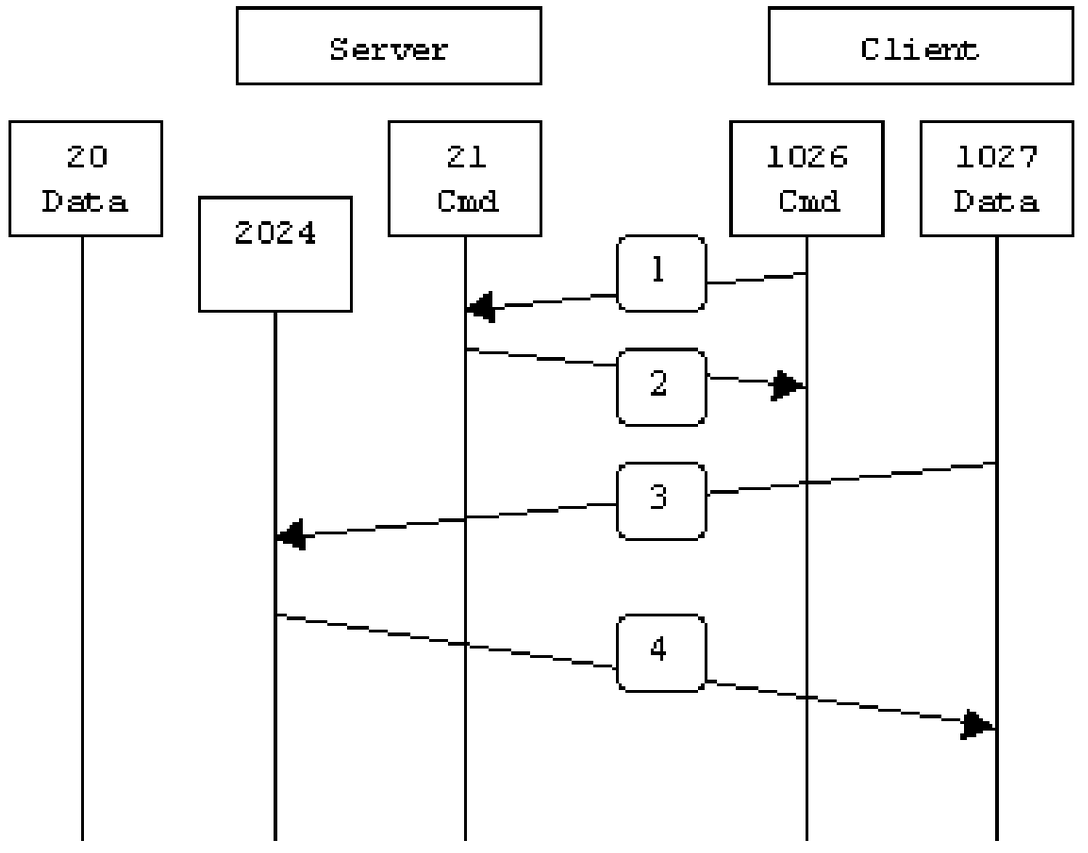
FTP

- In active mode FTP:
  - Client connects from a random unprivileged port (N > 1023) to the FTP server's command port, port 21.
  - Client starts listening to port N+1 and sends the FTP command PORT N+1 to the FTP server.
  - Server connects back to the client's specified data port from its local data port, which is port 20.
  - Problem?



FTP

- Passive mode FTP:
  - Client opens two random unprivileged ports locally (N > 1023 and N+1).
  - The first port contacts the server on port 21
  - Instead of then issuing a PORT command, client issues the PASV command.
  - Server opens a random unprivileged port (P > 1023) and sends the PORT P command back to the client.
  - Client then initiates the connection from port N+1 to port P on the server to transfer data.



Saving FTP (summary)

- By default, FTP clients send a PORT command to specify the address for an inbound connection
- If the PASV command is used instead, the data channel uses a separate outbound connection
- If local policy permits arbitrary outbound connections, this works well

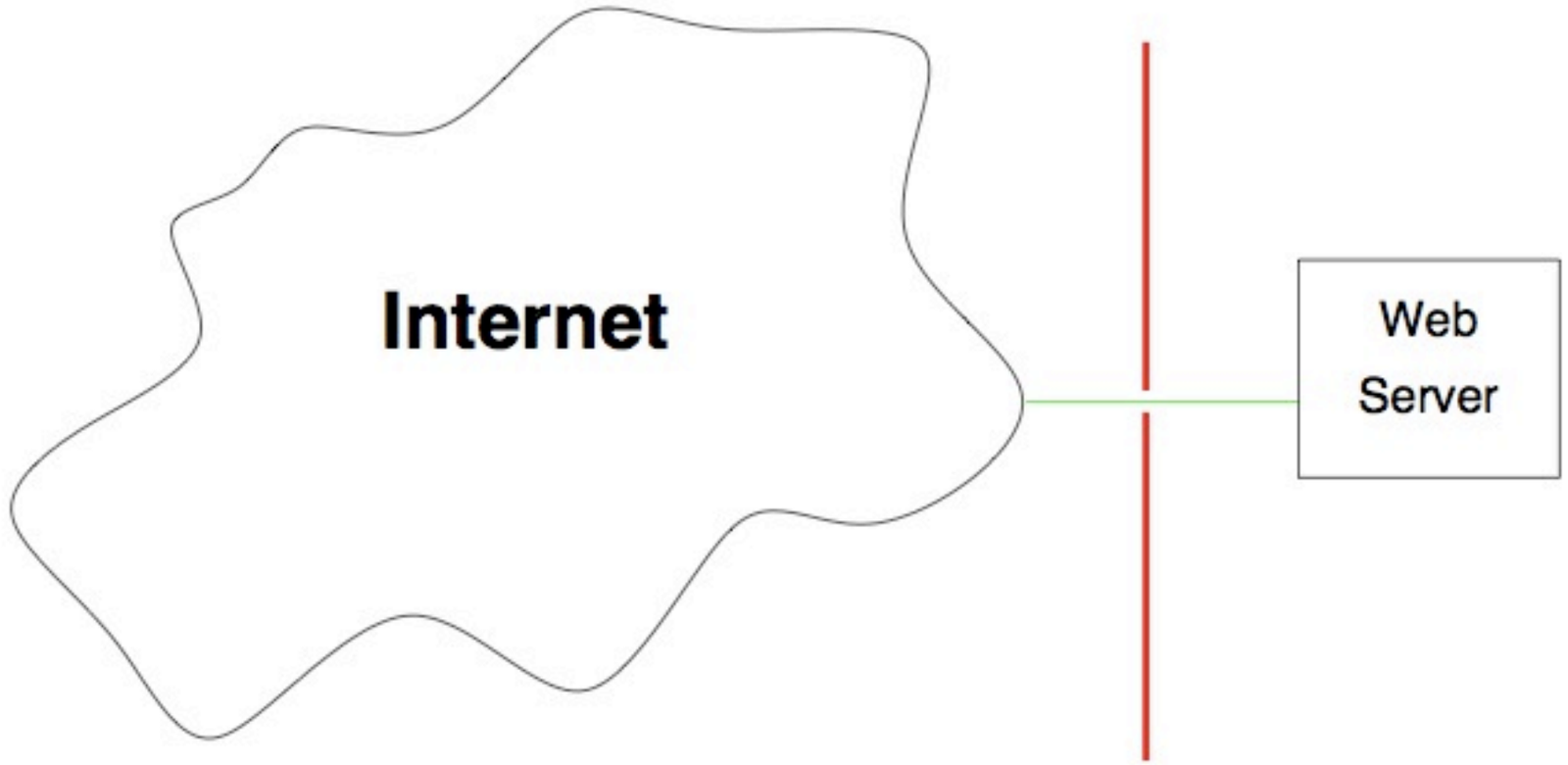
The Role of Packet Filters

- Packet filters are not very useful as general-purpose firewalls
- That said, they have their place
- Several special situations where they’re perfect

Simplicity

- Packet filters are very simple, and can protect some simple environments
- Virtually all routers have the facility built in

Point Firewalls

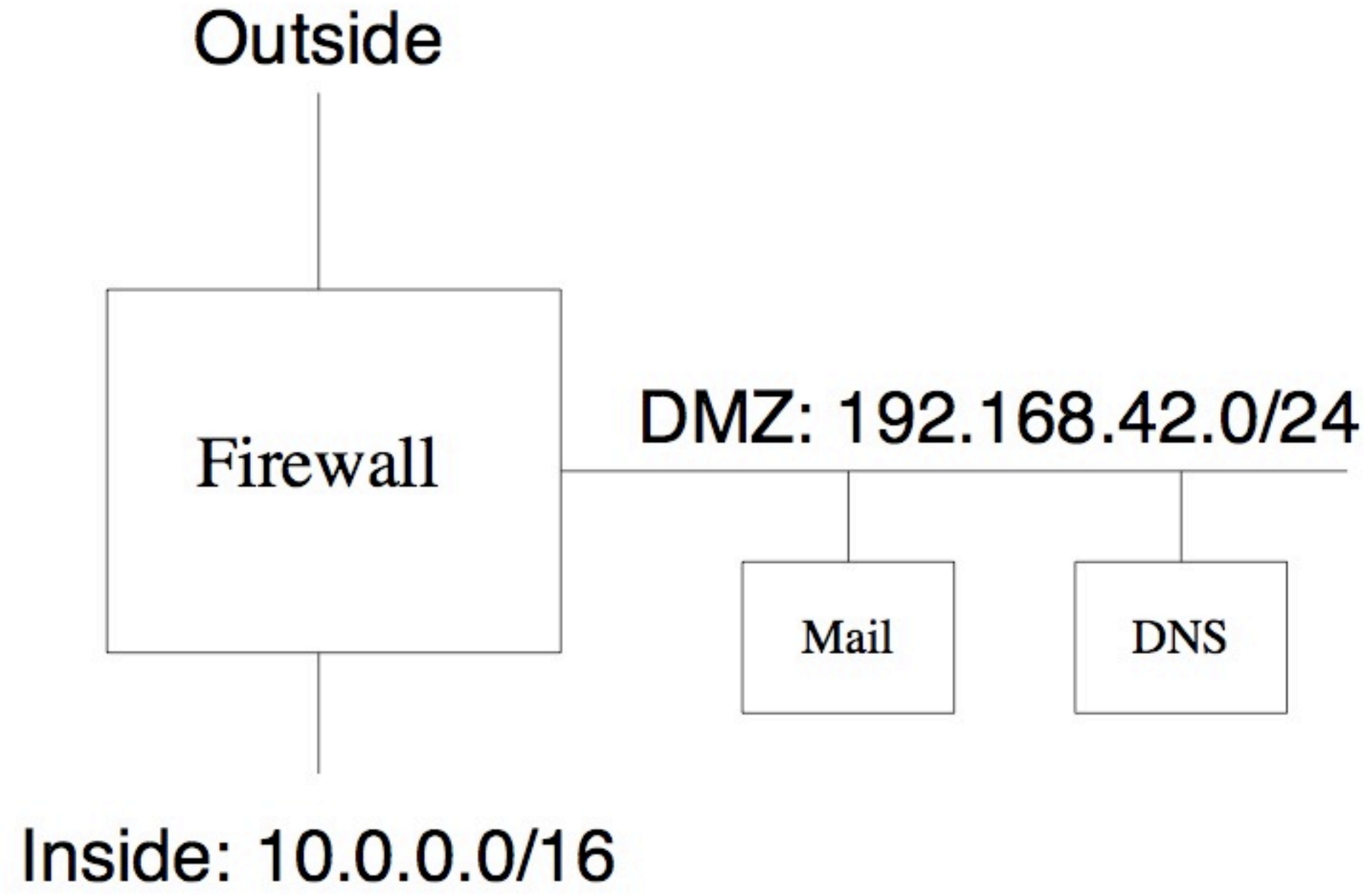


- Allow in ports 80 and 443. Block everything else. This is a Web server appliance — it shouldn't do anything else! But — it may have necessary internal services for site administration.

Address Filtering

- At the border, block internal addresses from coming in from the outside
- Similarly, prevent fake addresses from going out

Sample Configuration



Sample Rules

Interface	Action	Addr	Port	Flags
Outside	Block	src=10.0.0.0/16		
Outside	Block	src=192.168.42.0/24		
Outside	Allow	dst=Mail	25	
Outside	Block	dst=DNS	53	
Outside	Allow	dst=DNS	UDP	
Outside	Allow	Any		ACK
Outside	Block	Any		
DMZ	Block	src !=192.168.42.0/24		
DMZ	Allow	dst=10.0.0.0/16		ACK
DMZ	Block	dst=10.0.0.0/16		
DMZ	Allow	Any		
Inside	Block	src !=10.0.0.0/16		
Inside	Allow	dst=Mail	993	
Inside	Allow	dst=DNS	53	
Inside	Block	dst=192.168.42.0/24		
Inside	Allow	Any		

Stateful Packet Filters

Stateful Packet Filters

- Most common type of packet filter
- Solves many — but not all — of the problems with simple packet filters
- Requires per-connection state in the firewall

Keeping State

- When a packet is sent out, record that
- Associate inbound packet with state created by outbound packet

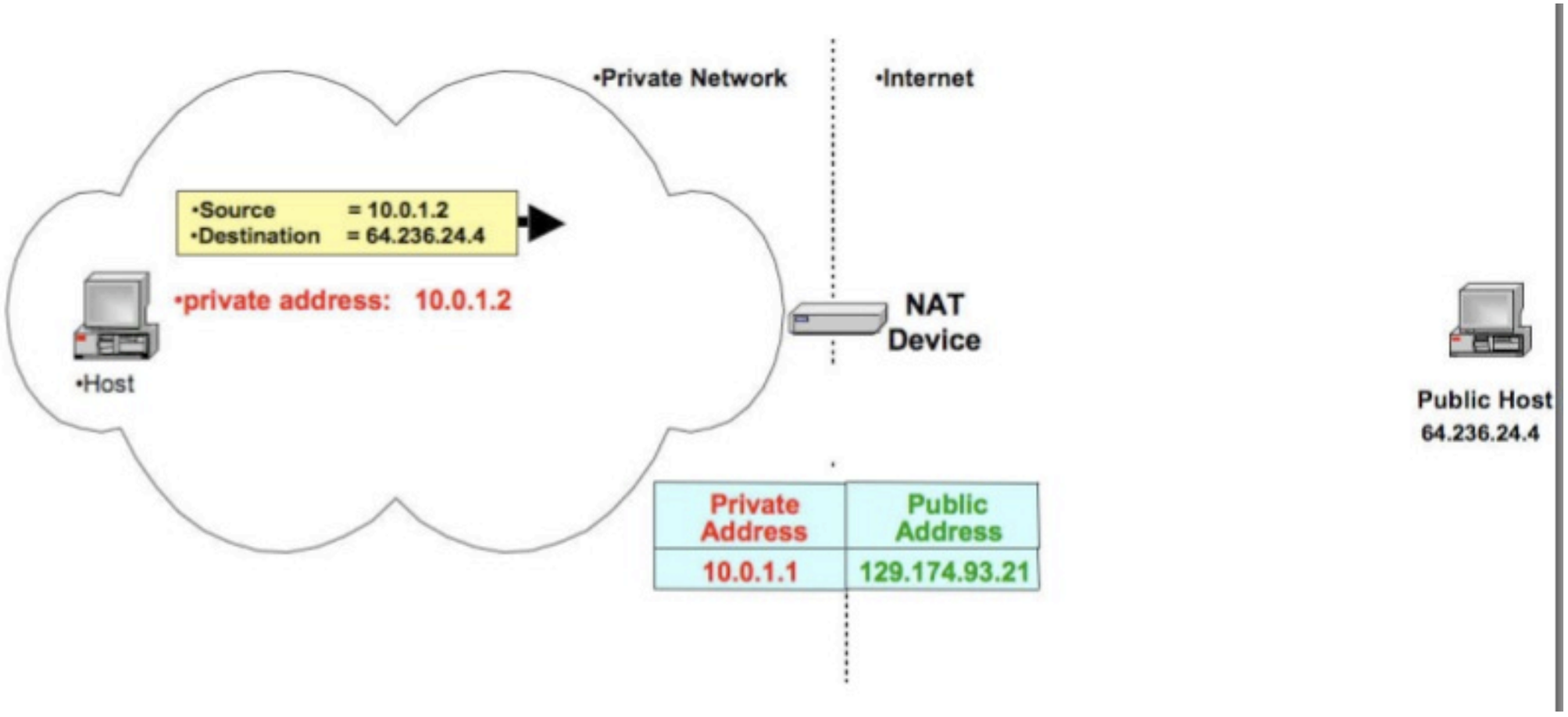
Problems Solved

- Can handle UDP query/response
- Can associate ICMP packets with connection
- Solves some of the inbound/outbound filtering issues — but state tables still need to be associated with inbound packets

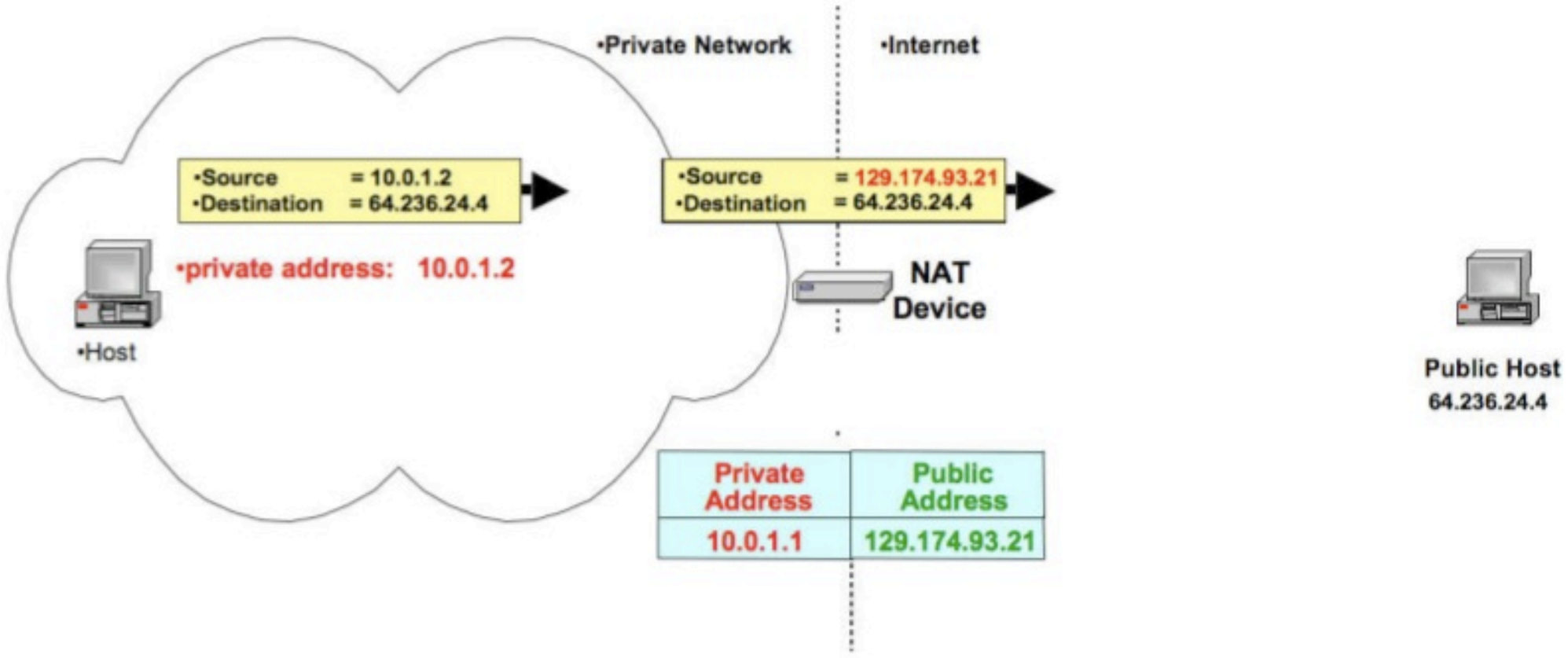
Network Address Translators

- Translates source address (and sometimes port numbers)
- Primary purpose: coping with limited number of global IP addresses
- Sometimes marketed as a very strong firewall — is it?
- It’s not really stronger than a stateful packet filter

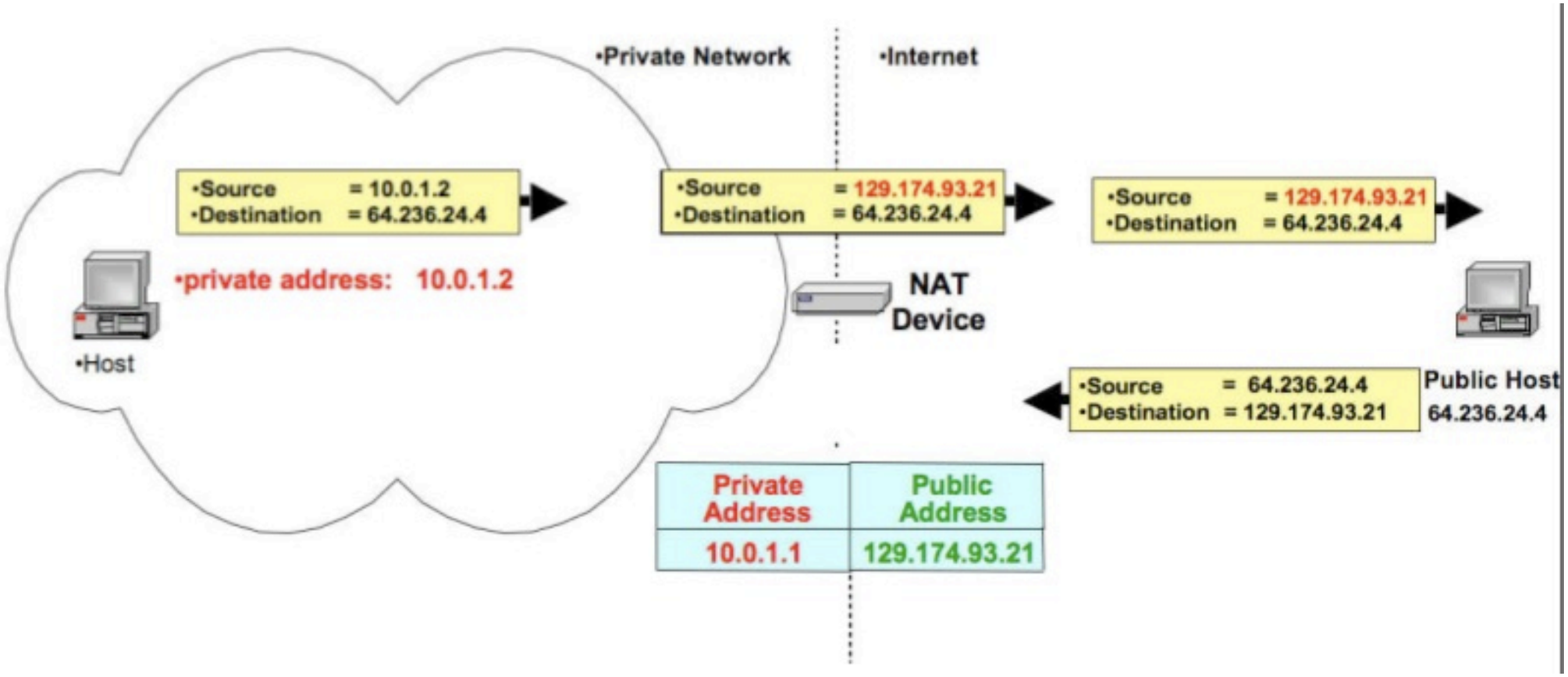
Basic NAT operation



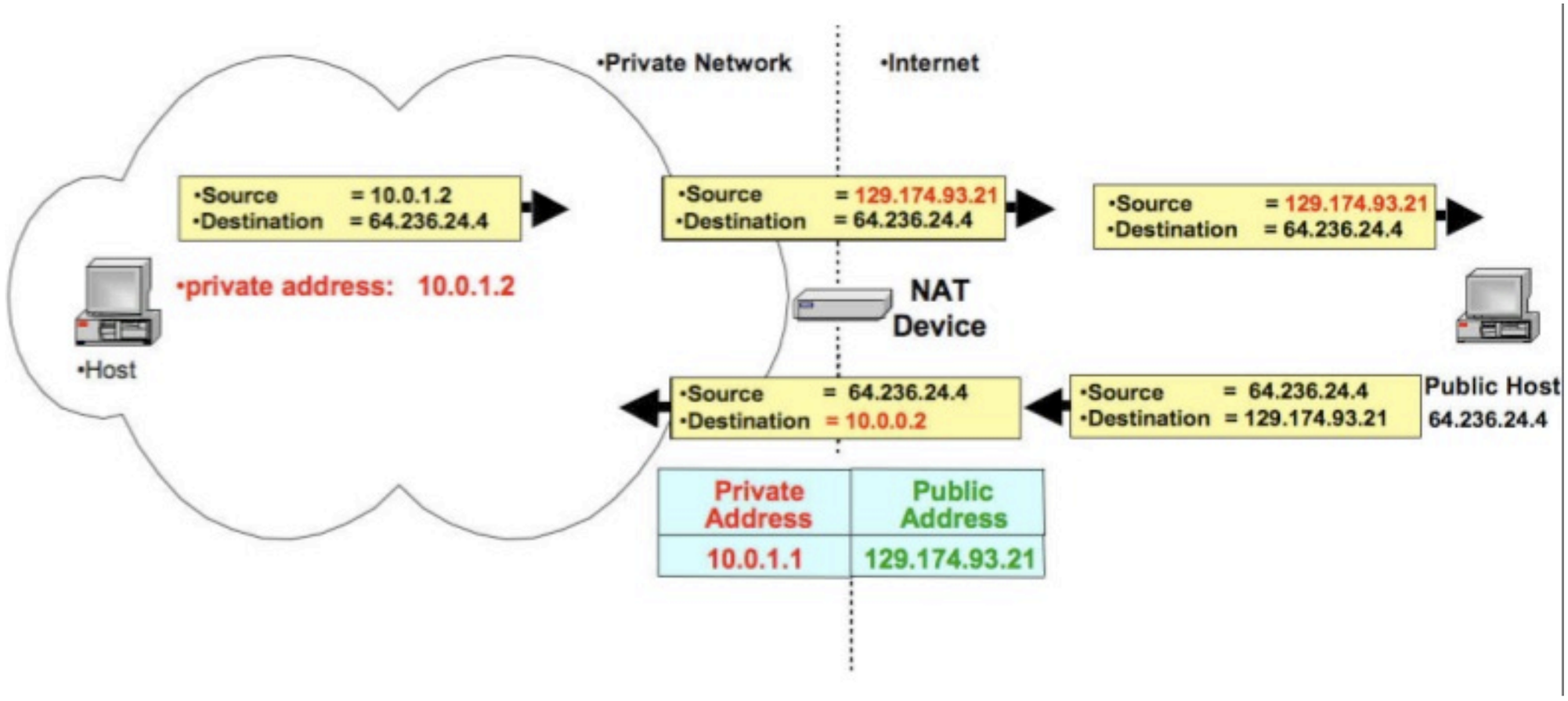
Basic NAT operation



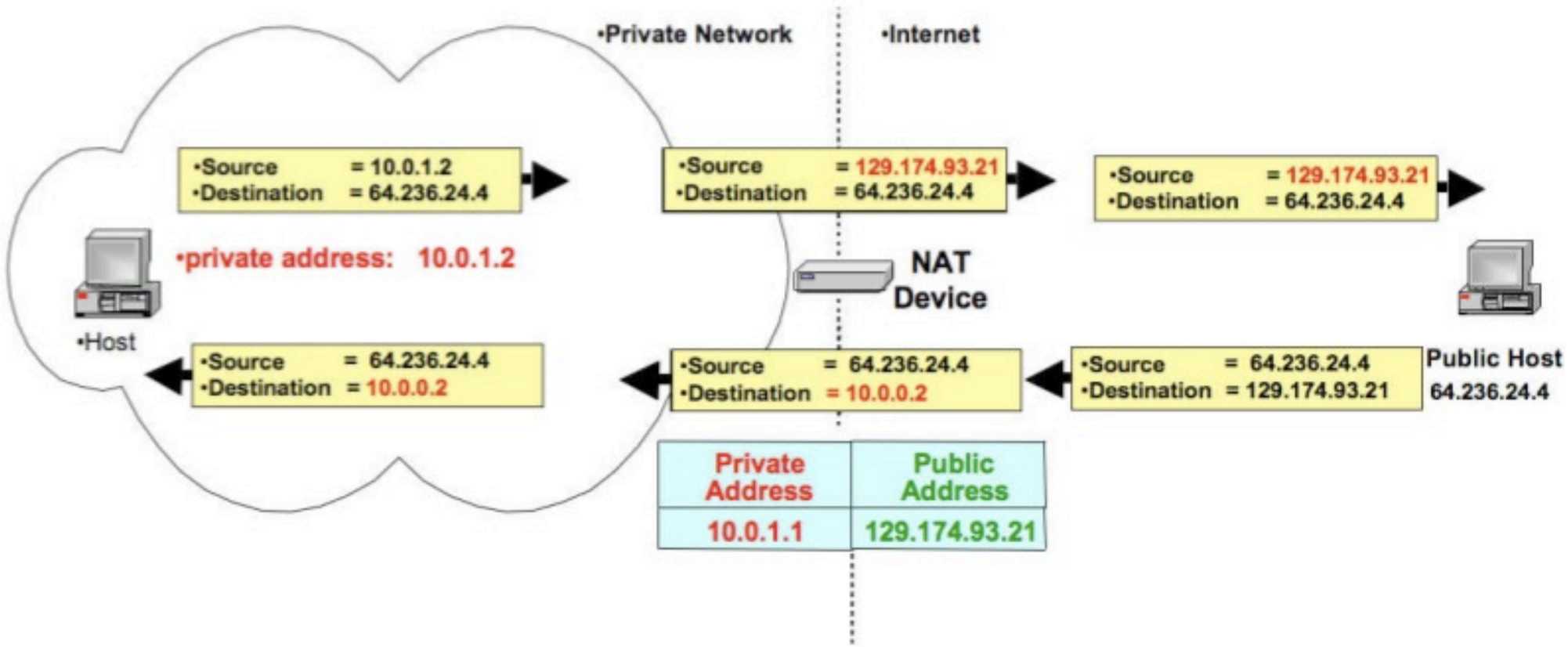
Basic NAT operation



Basic NAT operation



Basic NAT operation



Comparison

Stateful Packet Filter	NAT
Outbound Create state table entry.	**Outbound** Create state table entry. Translate address.
Inbound Look up state table entry; drop if not present	**Inbound** Look up state table entry; drop if not present. Translate address.

- The lookup phase and the decision to pass or drop the packet are identical; all that changes is whether or not addresses are translated.

Acknowledgments/References

- [Bellovin 06] COMS W4180 — Network Security Class Columbia University, Steven Bellovin, 2006.
- [Stavrou07] ISA 656, Network Security, Angelos Stavrou, Fall 2007. George Mason University.
- [Memon03] CS 392/682 – Network Security, Nasir Memon, Spring 2003. Polytechnic University.