Computer and Network Security: Firewalls

Kameswari Chebrolu

All the figures used as part of the slides are either self created or from the public domain with either 'creative commons' or 'public domain dedication' licensing. The public sites from which some of the figures have been picked include: http://commons.wikimedia.org (Wikipedia, Wikimedia and workbooks); http://www.sxc.hu and http://www.pixabay.com

Outline

- What are Firewalls?
- Firewall Theory
- Types of Firewalls
- Implementing Firewalls
- Circumventing Firewalls

Securing Networks

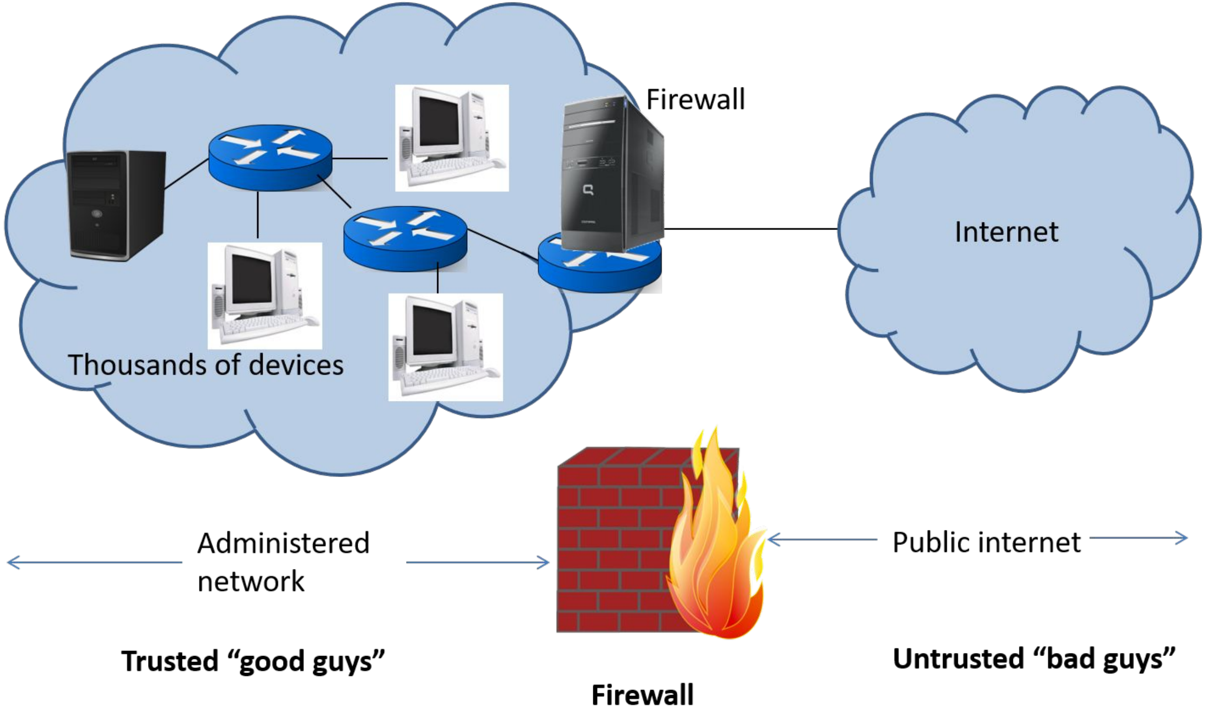
- Organization has many networked computing devices. How to protect them?
  - Very large surface area for possible attack
- How about defence mechanisms in each system?
  - Disable unused services, insist use of secure protocols etc
  - Challenge: Systems use different OS, hardware, provide different services
    - Complex Management, just does not scale
- How is it done in real-life?

Real Life Situation

- How is security provided in a large campus like IIT Bombay or a big mall with many shops?
  - Guard all entrances (check posts)
  - Check identity/bags of those entering and leaving at these check posts
- Firewalls do same in the networking world

Firewalls

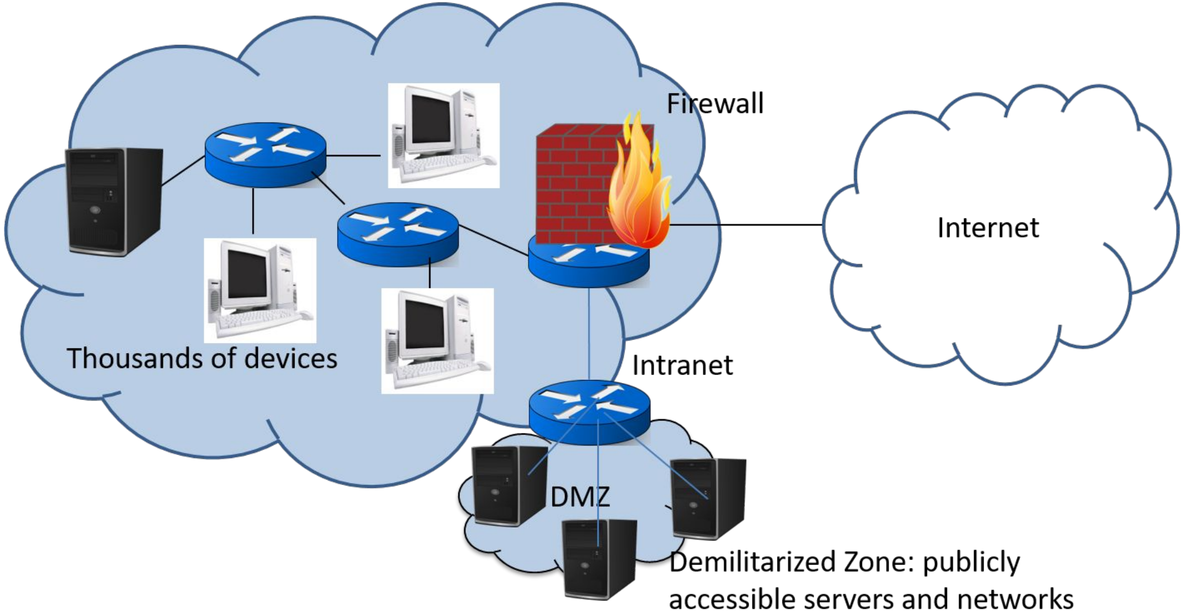
- Isolates the internal network from external Internet
- Implement a security policy



Policy

- Earlier: What **action principals** can take on an **object** (Only Bob may use this machine)
- Here: Who can talk with whom to get what service?
- Two types of connections:
  - Inbound: External users talk with internal users
  - Outbound: Internal users talk with external users

- Sample policy:
  - Insiders can access any outside service
  - Outsiders can access service only of machines in DMZ (demilitarized zone)



Demilitarized Zone: publicly accessible servers and networks

- Location of Firewall
  - Gateway of any sensitive network (like in previous figure)
  - Can also be at end-hosts

Traffic not captured by the policy?

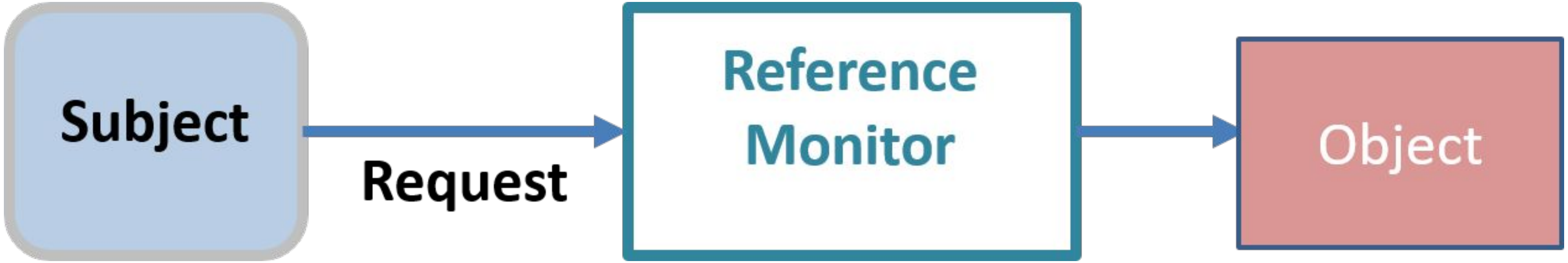
- Default Allow: Permit access to services
  - Disallow in case of a problem
  - Convenient (people more happy) but dangerous
- Default Deny: Don’t permit access to services
  - Allow when users complain; devise a specific policy
  - Less convenient (people less happy) but more secure
- Good practice: Default Deny
  - More secure and issues can be quickly identified

Outline

- What are Firewalls?
- **Firewall Theory**
- Types of Firewalls
- Implementing Firewalls
- Circumventing Firewalls

Reference Monitor

- A security concept
- Reference Monitor (RM) examines every request to a controlled resource (object)
- Decides whether to allow or deny the request



Security Properties

Need to ensure three properties

- Always invoked: Every access to the resource is mediated by RM
- Tamper Resistant: Integrity of RM always maintained
  - No code or state change
- Verifiable: Verify RM is doing its job
  - RM needs to be simple to verify this

Firewalls as RM

1. Always Invoked?
   - Firewalls implemented at chokepoints check all incoming and outgoing traffic

- But what about?
  - A user setting up an insecure Wireless AP within organization
  - A user connecting an infected machine to the network
- Need to cover all links
  - These set of links determine security perimeter
  - Difficult to achieve in practice

2. Tamper Resistant?

- Feasible. How?
  - Allow access to firewall machine via stringent authentication mechanisms
  - Physically protect firewall

3. Verifiable?

- Tough in practice when the number of rules are large

Outline

- What are Firewalls?
- Firewall Theory
- **Types of Firewalls**
- Implementing Firewalls
- Circumventing Firewalls

Types of Firewalls

- Stateless Packet Filters
- Stateful Packet Filters
- Application Gateways

Stateless Packet Filters

- Implemented on routers via Access Control Rules
  - List of these rules is called ACL (Access Control List)
  - Different ACLs for each router interface
- Firewall checks each packets individually (hence no state) against rules
  - Only looks at packet headers: Layer 3, Layer 4 headers
    - E.g. Source IP, destination IP, source port, destination port, TCP flags, Packet type (e.g. ICMP), wild cards
  - Rules specify action (allow or drop) against a matching packet
  - Rules are applied top to bottom
    - Go to next rule only if the current rules does not match

Examples

- Only an external client at 12.7.8.9 on port 5000 can connect to a special web service set up within your organization on 21.3.5.6

Action	Src IP	Dst IP	Protocol	Src Port	Dst Port	TCP flags
Allow	12.7.8.9	21.3.5.6	TCP	5000	80	-
Deny	*	21.3.5.6	TCP	*	80	-

Even packets from 12.7.8.9 on any other port will be dropped

Examples

Action	Src IP	Dst IP	Protocol	Src Port	Dst Port	TCP flags
Allow	12.7.8.9	21.3.5.6	TCP	5000	80	-
Deny	*	21.3.5.6	TCP	5000	80	-

vs

Action	Src IP	Dst IP	Protocol	Src Port	Dst Port	TCP flags
Deny	*	21.3.5.6	TCP	5000	80	-
Allow	12.7.8.9	21.3.5.6	TCP	5000	80	-

External client at 12.7.8.9 on port 5000 cannot connect to a special web service any more

Order Matters!

Another Example

- Organization Policy: Internal users can surf the web; block every thing else □ permit DNS traffic for URL resolutions
  - No connections from outside to inside are allowed
  - But external web traffic corresponding to internal user requests needs to get in
- Organization address: 125.5 / 16

ACL

action	source address	dest address	protocol	source port	dest port	flag bit
allow	125.5/16	outside	TCP	any	80	any
allow	outside	125.5/16	TCP	80	any	ACK
allow	125.5/16	outside	UDP	any	53	---
allow	outside	125.5/16	UDP	53	any	----
deny	all	all	all	all	all	all

- First two rules: Internal users can surf web
  - A TCP connection establishment from outside to inside will have syn bit set, which will be dropped
- Second two rules: Allow DNS traffic to flow

Points to Note

- An organization can have 1000s of such rules
  - Easy to introduce bugs which attackers can exploit
- Systematic evaluation is tough at scale

- Stateless: Can admit dangerous packets

action	source address	dest address	protocol	source port	dest port	flag bit
allow	125.5/16	outside	TCP	any	80	any
allow	outside	125.5/16	TCP	80	any	ACK

  - No TCP connection, but can admit some ACK packets related to it
  - IP Fragmentation attack:
    - First fragment: offset 0, the TCP header has only ack bit set
    - Second fragment: offset X □ overlapping data
      - Not examined by Firewall since it is a second fragment
      - During reassembly, this data overwrites first fragment □ syn bit set, ack bit not set

Stateful Firewalls

- Most firewalls are of this type
- At establishment of connection, make a decision whether to admit or not
  - Any later packet not part of admitted connections are dropped
- Example: TCP
  - Track SYN/FIN; timer to prune inactive connections
  - (in prev example) Packet with just ack bit set will not be admitted
- Drawback: Memory; Can slow down connections

ACL

action	source address	dest address	protocol	source port	dest port	flag bit	Conn check
allow	125.5/16	outside	TCP	any	80	any	
allow	outside	125.5/16	TCP	80	any	ACK	X
allow	125.5/16	outside	UDP	any	53	---	
allow	outside	222.22/16	UDP	53	any	----	X
deny	all	all	all	all	all	all	

Connection Table:

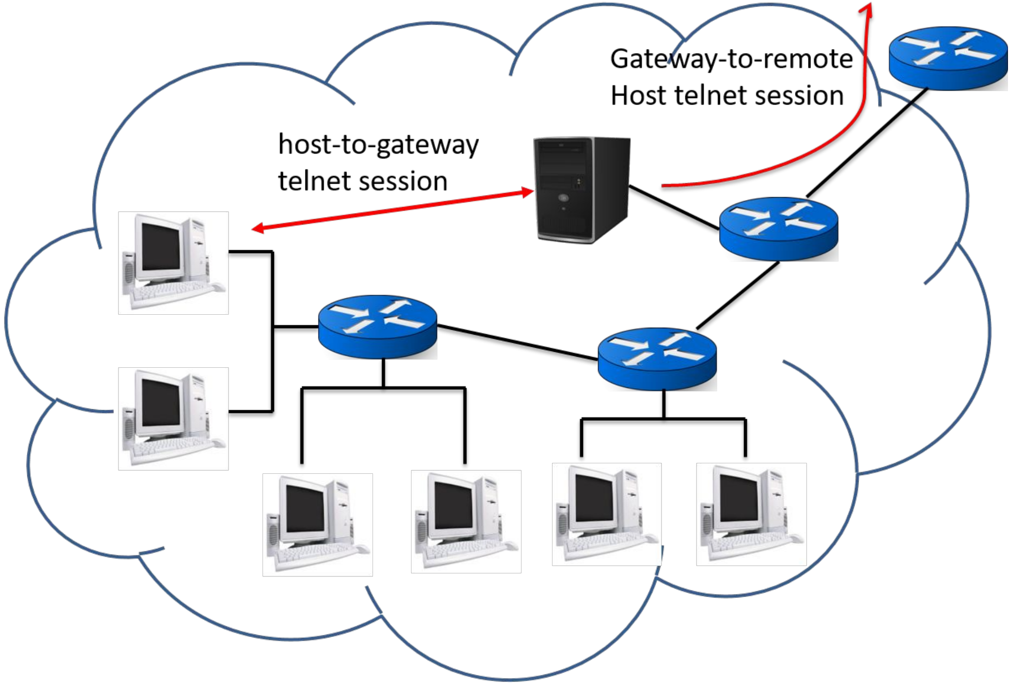
Source Address	Source Port	Destination Address	Destination Port
125.5.12.14	4533	120.12.3.1	80
125.5.19.34	6771	12.14.5.6	80

Example

- Block all telnet connections to the outside world
- But permit a few select users to telnet outside
- How about user IP in the ACL?
  - IP spoofing issues
  - User may want to telnet from any machine
- How achieved?
  - Need to look at application data

Application Gateway

- Users telnet to gateway
- Gateway authenticates the user (e.g. passwd based)
- Gateway telnets to destination
  - Gateway acts as a relay
- Firewall ACL permits telnet connections only from gateway


Gateway-to-remote Host telnet session
host-to-gateway telnet session

Drawbacks of Application Gateways

- Different applications need different gateways
- Client should know which gateway to connect to

Personal Firewalls

- Saw how firewalls protect networks
- Firewalls can protect personal machines too!
  - User defines ACL rules; checked against all incoming and outgoing packets
  - Collect logs to monitor and debug
  - Combine with virus scanners for better security

Firewall Drawbacks

- Interfere with some applications (e.g. Skype)
- Don’t solve all problems
  - Server vulnerabilities can be exploited (SQL injection, buffer overflow)
  - Protocol implementations can be exploited
  - Most DDOS attacks cannot be prevented
  - Insider attacks cannot be prevented
- More rules/misconfiguration □ susceptible to attacks
- Can only prevent “known” attacks

Outline

- What are Firewalls?
- Firewall Theory
- Types of Firewalls
- Implementing Firewalls
- Circumventing Firewalls

Firewall Implementation (in Linux)

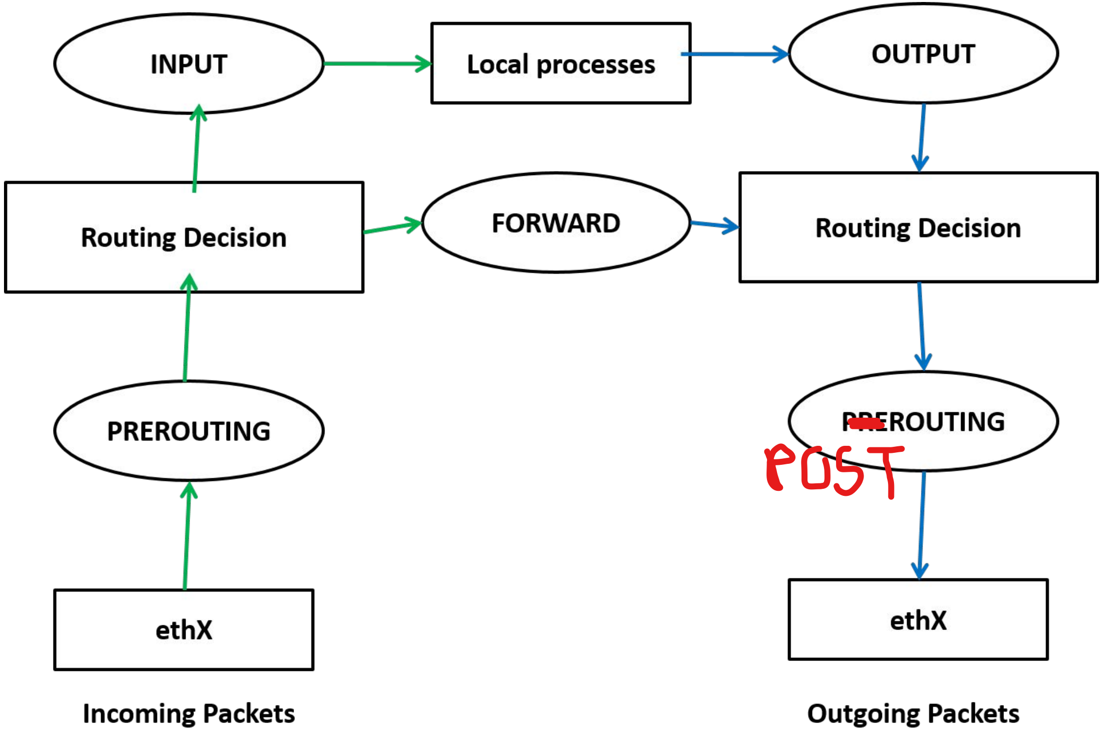
- Netfilter hooks (kernel’s packet filtering framework)
- IPTables (user level firewall tool)

Firewall Implementation

- IP packet processing happens at kernel level
- How to modify processing to implement firewall functionality?
  - Kernel level code changes difficult
- Newer kernels provide hooks at several points of packet processing □ netfilter hooks
- Can write kernel modules that register with these hooks and get packets to process
  - Still not so easy

NetFilter Hooks

- 5 hooks provided by kernel (oval boxes)



IPTables

- Permit operation at user-space
  - Program built on top of netfilter hooks
- Uses Tables to organize rules
  - Rule related with NAT put in NAT table
  - Rule related to allow/deny packets put in Filter table
- 5 Tables:
  - Filter: filters packet
  - NAT: Nat related functionality
  - Mangle: alters IP headers (e.g. TTL)
  - RAW: mark packets to opt out of connection tracking
  - Security: SElinux related functions

IPTables

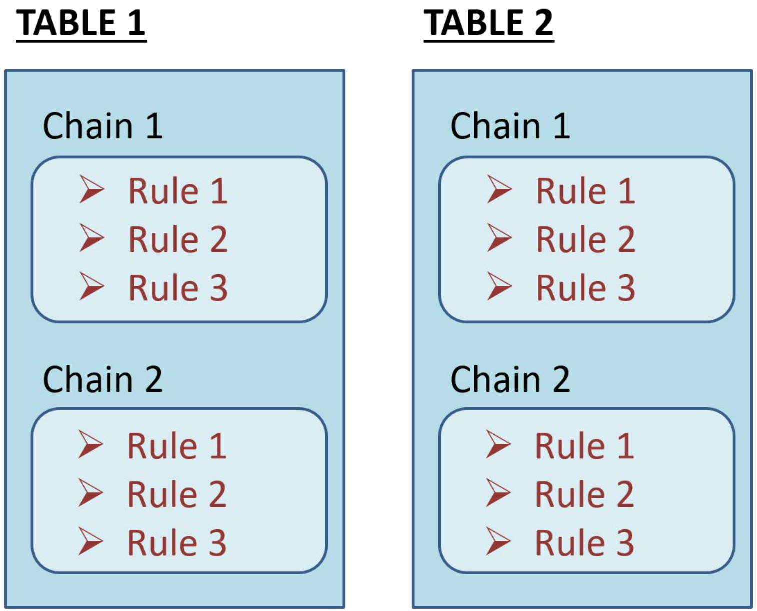


TABLE 1

Chain 1

- Rule 1
- Rule 2
- Rule 3

Chain 2

- Rule 1
- Rule 2
- Rule 3

TABLE 2

Chain 1

- Rule 1
- Rule 2
- Rule 3

Chain 2

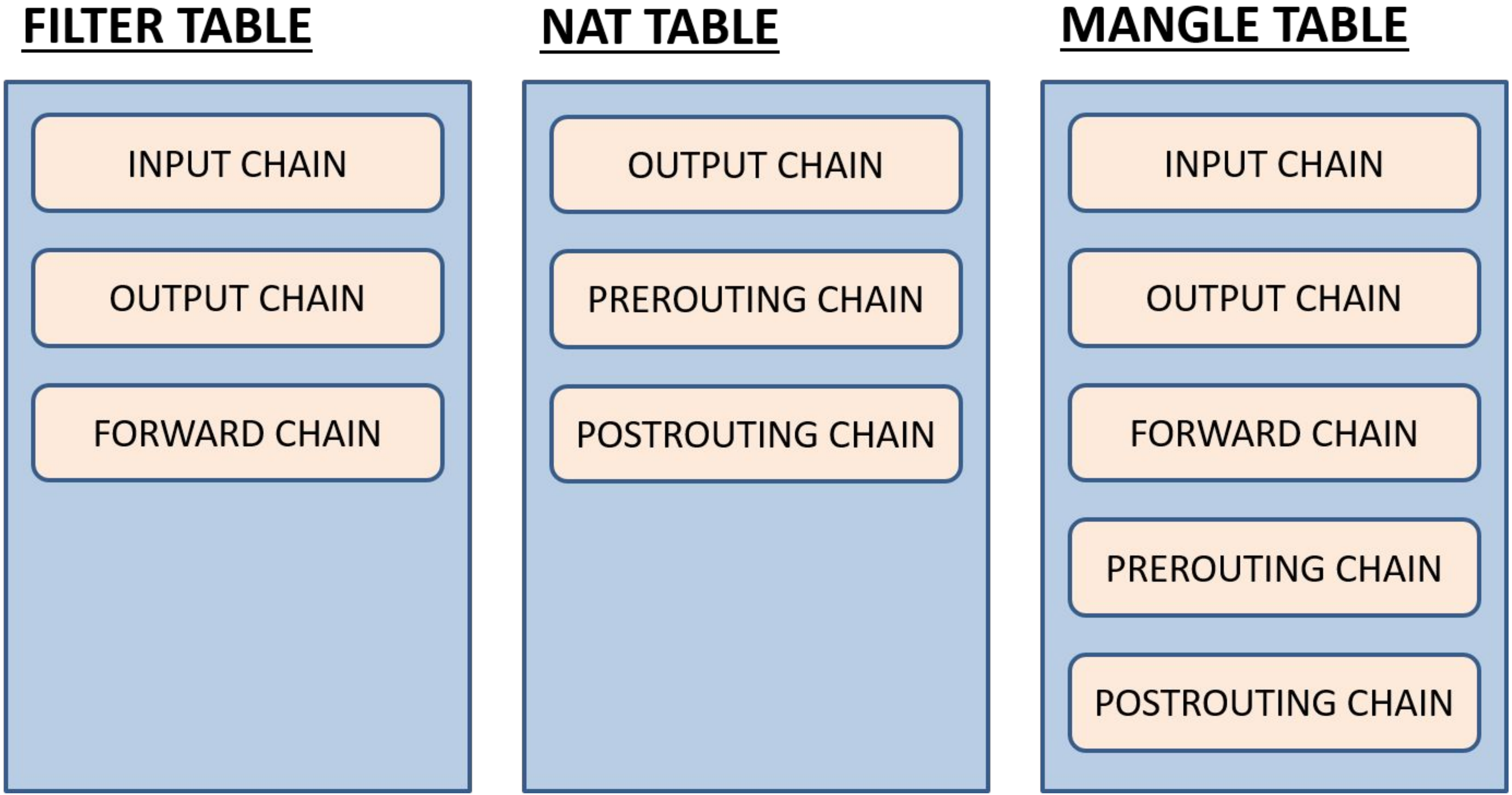
- Rule 1
- Rule 2
- Rule 3

Uses tables to organize firewall rules

IP tables is a bunch of Tables (tables represent a type of action; e.g. Filter, Nat etc)

Tables are a bunch of chains (chains represent netfilter hooks, e.g. Input, Pre-routing etc)

Chains are a bunch of firewall rules



FILTER TABLE

- INPUT CHAIN
- OUTPUT CHAIN
- FORWARD CHAIN

NAT TABLE

- OUTPUT CHAIN
- PREROUTING CHAIN
- POSTROUTING CHAIN

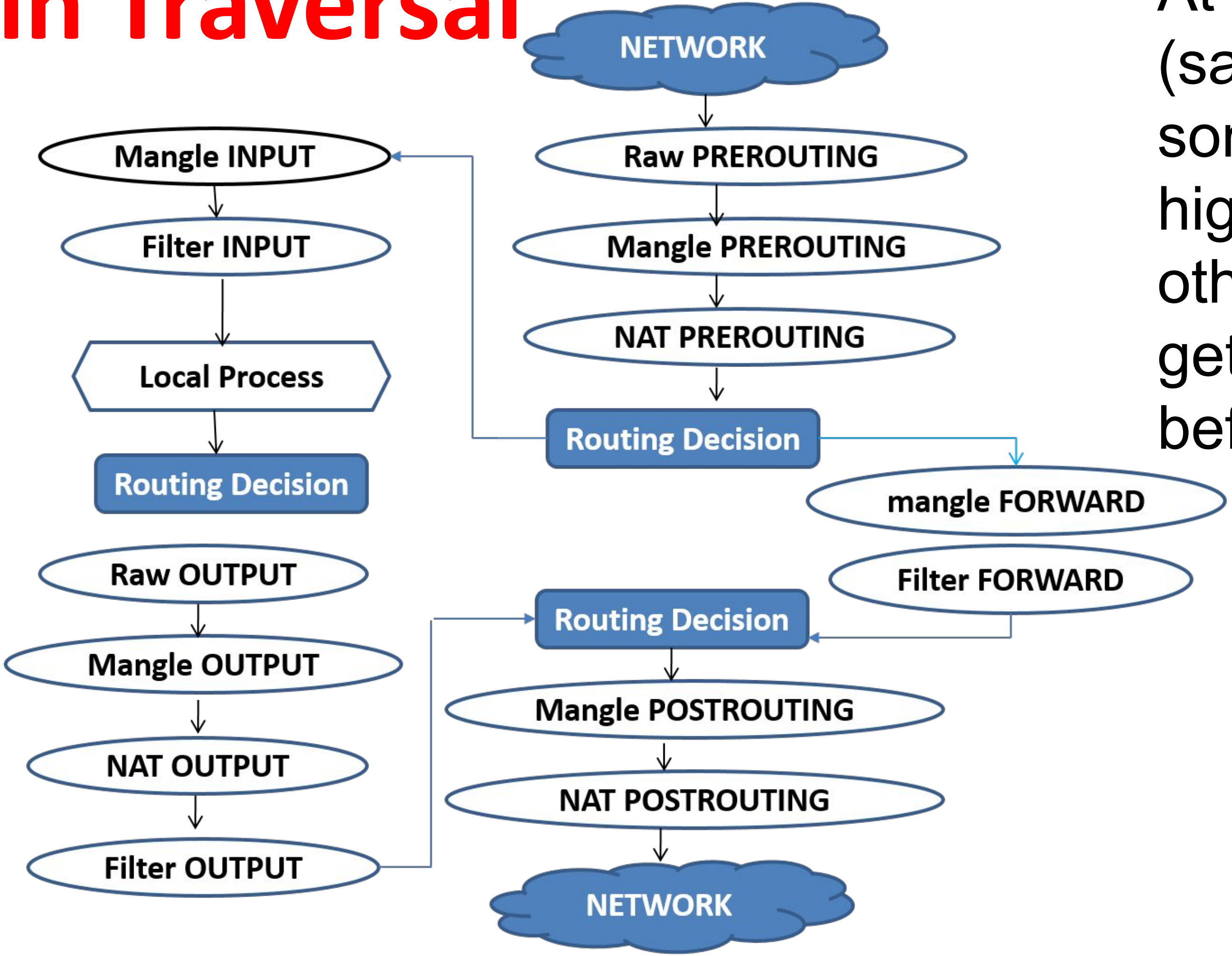
MANGLE TABLE

- INPUT CHAIN
- OUTPUT CHAIN
- FORWARD CHAIN
- PREROUTING CHAIN
- POSTROUTING CHAIN

Tables/Chains→	PREROUTING	INPUT	FORWARD	OUTPUT	POSTROUTING
raw	✓			✓	
mangle	✓	✓	✓	✓	✓
nat (DNAT)	✓			✓	
filter		✓	✓	✓	
security		✓	✓	✓	
nat (SNAT)		✓			✓

Not all tables used at every hook
At a hook, tablets are processed in the above order
(top to bottom; e.g raw > mangle > nat)

Chain Traversal



At a given hook (say Pre-routing), some tables have higher priority ove others (e.g. raw gets handled before mangle)

Rules

- Rules have a matching component and a target
  - When matching criteria met, target is executed
  - When matching criteria not met, move to next rule
- Target: accept, drop, queue, return
- Example: iptables –t filter –A OUTPUT –p tcp --dport 80 –j drop
  - Filter table, OUTPUT chain, match: tcp protocol, with destination port as 80, target: drop
  - You cannot access HTTP from the machine

Outline

- ~~What are Firewalls?~~
- ~~Types of Firewalls~~
- ~~Implementing Firewalls~~
- Circumventing Firewalls

Circumventing Firewalls: Inside to Outside

University setting:

- Students spending lot of time gaming (external server) an not studying
- Policy: Block traffic to this service
- Suppose the service runs on port 7777
- Firewall rule in the university

Policy	Src.addr	Src.port	Protocol	Dst.addr	Dst.port
Deny	*	*	UDP	*	7777

- Gaming server losing traffic. How can they get around this?

Solution

- Move service to port 53 (DNS)
  - There is nothing binding a port to a service (arose out of convenience in locating services)
  - Client / server need to agree on the ports
- Can the university deny traffic of this port?
  - No since legitimate DNS traffic will also be dropped

Twist to the problem

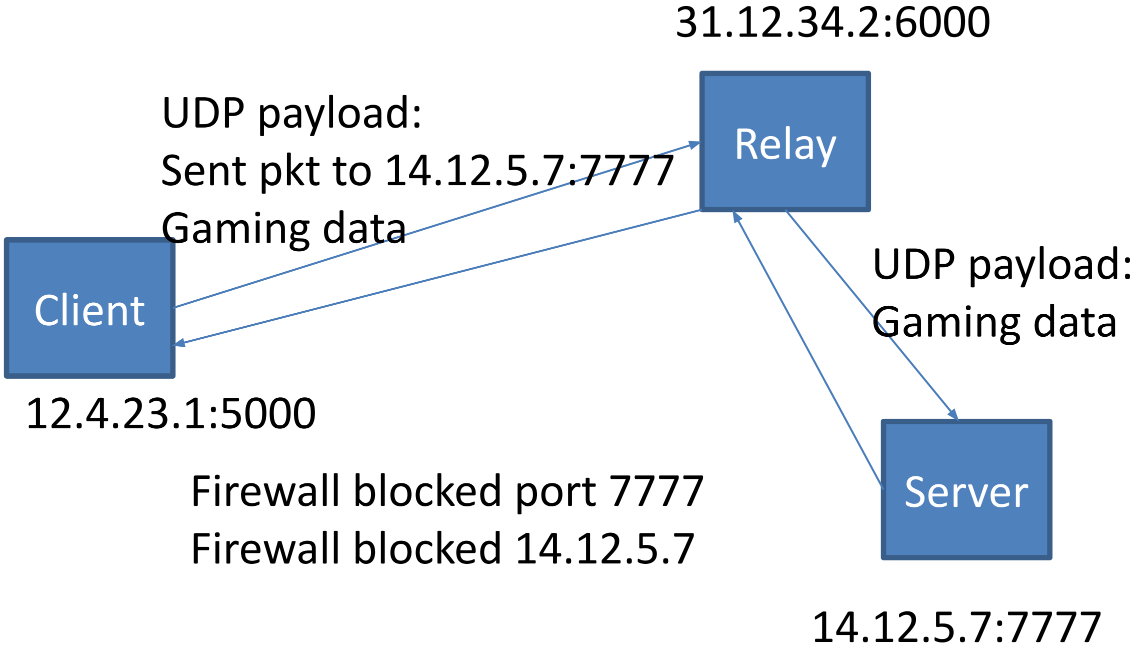
- What if the gaming server not interested in changing port? How can a student still access the service?

(or)

- What if the university blocked the IP address of the gaming server?

Relay based Solution

- Use a relay
- Firewall will allow relay traffic
  - If it is blocked, moved to another relay



Generic Solution: Tunneling

- Allows a foreign protocol to run over a network that does not support it
  - E.g. IPv6 over IPv4 networks
- Based on encapsulation (encapsulate one protocol inside another)
  - Previous example: UDP within UDP
  - Another example: IP within SMTP
    - Use an IP packet as an email attachment;
    - End point decapsulates and acts on it
- Inner protocol cannot bypass firewall; Outer protocol can bypass

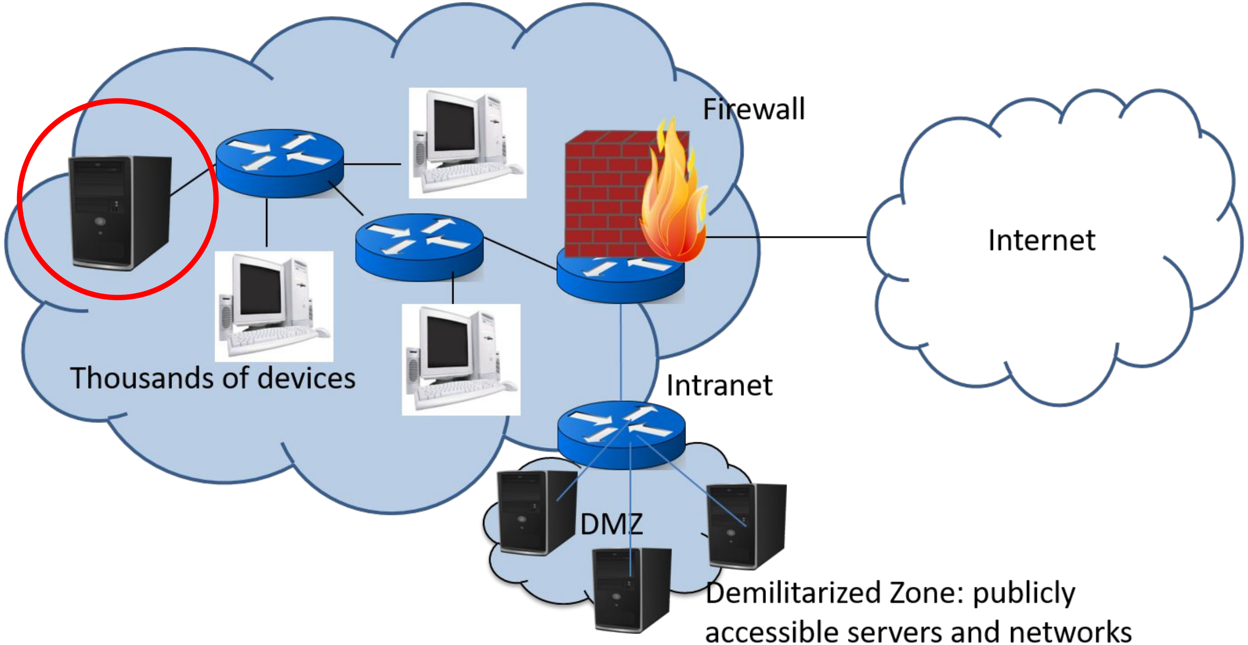
Circumventing Firewalls: Outside to Inside

How can an outside attacker sneak in?

- Figure out some flaw in the firewall
- Need some insider client support
- Not so easy!

Outside to Inside: Allowing Valid Users

- Firewall does not allow outsiders to access machines in intranet
- How to provide access for a genuine employee who is traveling?

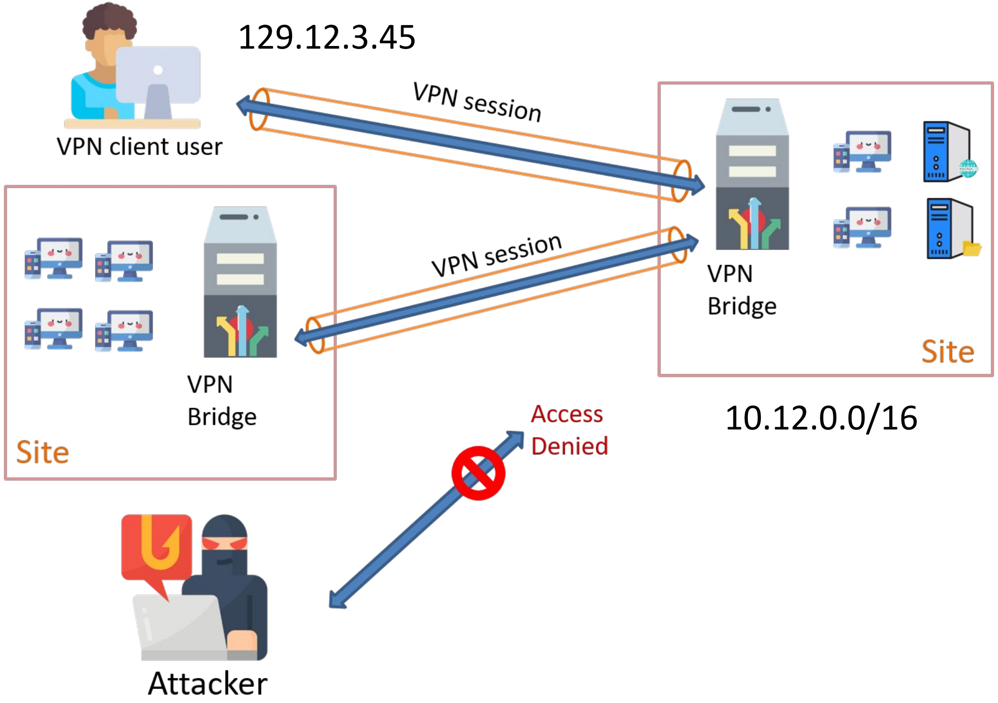

Firewall
Internet
Thousands of devices
Intranet
DMZ
Demilitarized Zone: publicly accessible servers and networks

Virtual Private Networks (VPNs)

- Based on Tunneling
  - VPN server acts as a relay
  - Outer header is directed to VPN server
  - Inner header appears as if VPN client is in local LAN
- Authentication, confidentiality, integrity handled by the tunnel

Inner header src: 10.12.11.7; dst: Local service

Outer header src: 129.12.3.45; dst: VPN server



Summary

- Firewalls provide perimeter security but are not fool proof
- Three types of firewalls: stateless, stateful and application gateway
- Implementation in Linux
  - netfilter hooks (kernel space) and iptables at user space
- Tunneling can circumvent firewalls for illegitimate (gaming) and legitimate use (VPNs)