Firewall

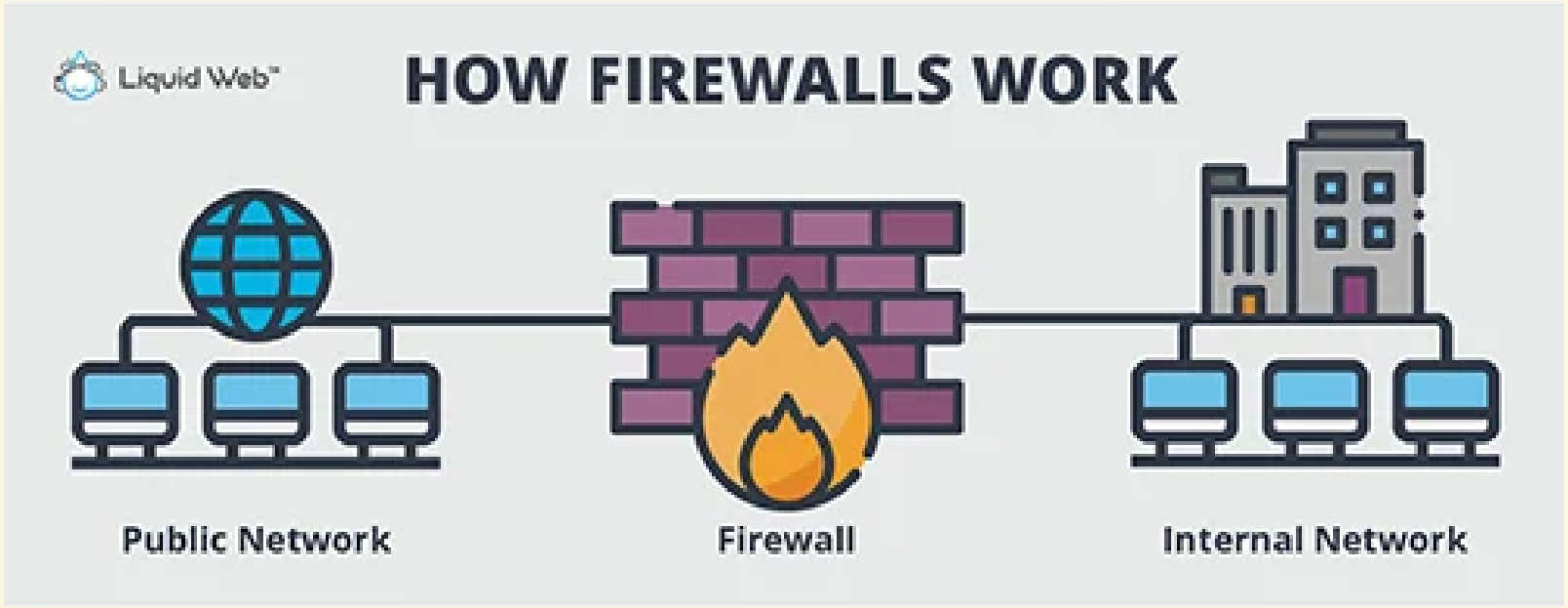
- CEG 6430/4430 Cyber Network Security
- Junjie Zhang
- junjie.zhang@wright.edu
- Wright State University

Firewall

A firewall intends to stop unauthorized network traffic.

- It can enforce policies to mediate network traffic.
- It needs to be immune to attacks by itself.

A firewall is typically deployed at the edge of the network but it can also be deployed at the host.


Liquid Web™
HOW FIREWALLS WORK
Public Network
Firewall
Internal Network

Firewall Takes Actions on Properties

- Properties
  - packet-level: IPs, ports, flags, and etc.
  - connection-level: establishment, directions, and etc.
  - application-level: states, keywords, and etc.
- Actions
  - Accept: allow the packet to go through
  - Deny: discard the packet
  - Reject: discard the packet and notify its source

Layers

- **Layer 3 Firewall**: act on IP-level information.
- **Layer 4 Firewall**: act on IP- and TCP/UDP-level information.
- **Layer 7 Firewall**: act on IP-, TCP/UDP-, and application-level information.

Statefulness

- A **stateless firewall** makes decision based on each individual packet.
- A **stateful firewall** makes decision based on a packet and the network session it blongs to.
- Stateless vs Stateful
  - Stateful offers more information to make the decision.
  - Stateless is more efficient does not need to keep connection states.

An Example

[Internal Network]------Firewall------[Internet]

To allow an internal host (130.108.0.0/16) to complete the 3-way handshake and then communicate with a web server with port 80 on Internet.

Stateless Firewall in Place

- TCP srcIP = 130.108.0.0/16, dstIP = *, srcPort = *, dstPort = 80, SYN, allow
- TCP srcIP = *, dstIP = 130.108.0.0/16, srcPort = 80, dstPort = *, SYN-ACK, allow
- TCP srcIP = 130.108.0.0/16, dstIP = *, srcPort = *, dstPort = 80, ACK, allow
- TCP srcIP = *, dstIP = 130.108.0.0/16, srcPort = 80, dstPort = *, ACK, allow

Stateless Firewall in Place

If an external host only sends an SYN-ACK or an ACK packet to 130.108.1.5:80?

- Will it go through the firewall?
- What is a possible reaction from 130.108.1.5? Does it leak any information?
- Should it be allowed?
- Can you write new rules to stop such packet?

Statefull Firewall in Place

- TCP srcIP = 130.108.0.0/16, dstIP = *, srcPort = *, dstPort = 80, SYN, allow
- TCP srcIP = *, dstIP = 130.108.0.0/16, srcPort = 80, dstPort = *, SYN-ACK, **established**, allow
- TCP srcIP = 130.108.0.0/16, dstIP = *, srcPort = *, dstPort = 80, ACK, allow
- TCP srcIP = *, dstIP = 130.108.0.0/16, srcPort = 80, dstPort = *, ACK, **established**, allow

Stateless Firewall in Place

If an external host only sends an SYN-ACK or an ACK packet to 130.108.1.5:80?

- Will it go through the firewall now?

iptables

iptables is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel.

- Works for Linux
- Can be used to implement a firewall

`iptables`

- *Rule*: the action(s) based on a packet.
- *Chain*: the location to enforce a rule.
- *Table*: a list of similar rules.

Four Default Tables

- Filter: to perform packet filtering.
- Network Address Translation (NAT): to modify source or destination address.
- Mangle: to adjust the IP header properties.
- Raw: to exempt packets from connection tracking. (A relatively new table.)

Chains

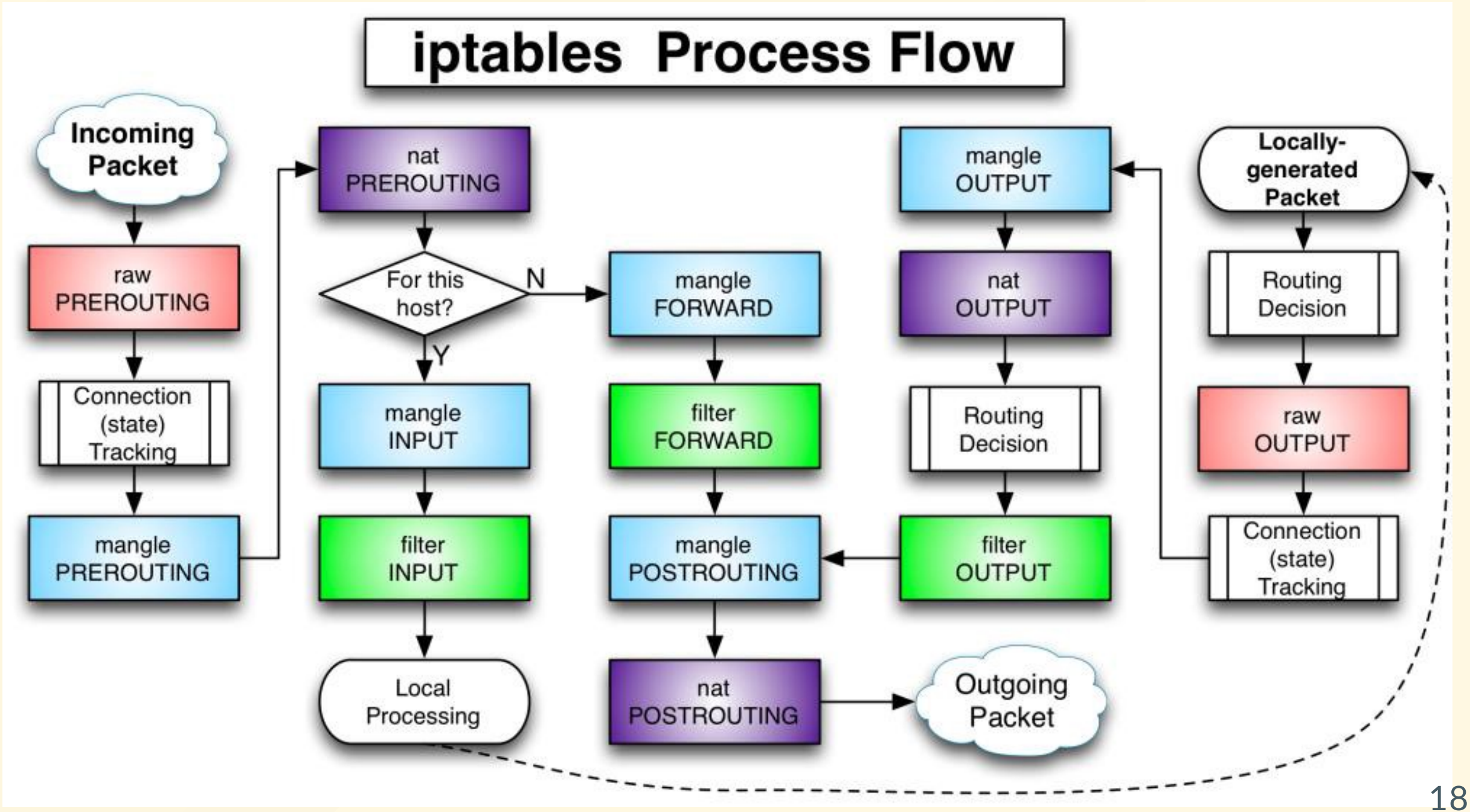
- Prerouting: This chain assigns a packet as soon as the packet arrives at the network interface, even if the packet does not destinate for this host.
- Input: This chain assigns a packet when a received packet actually destinates for this host.
- Forward: This chain assigns a packet when a received packet does not destinate for this host.

Chains (cont.)

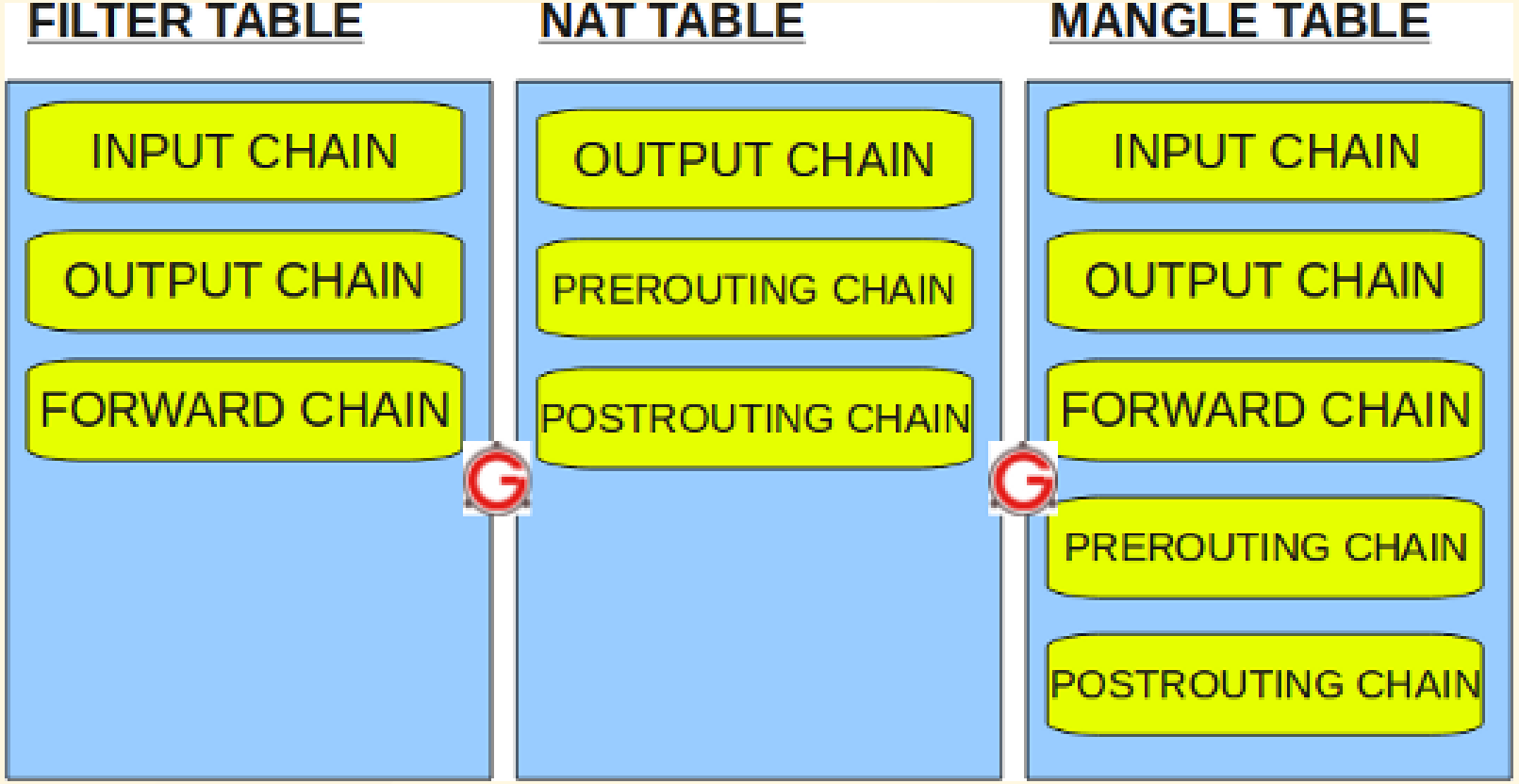
- Postrouting: This chain assigns a packet when a packet is after the routing decision.
- Output: This chain assigns a packet when a to-be-sent packet is created.

Tables and Chains

- A table can only contain a portion of chains.
- **Filter** is the default table and our focus. When the table name is not specified, a rule will be added to the **Filter** table.
- **Table** and **Chain** together decide where a rule will be enforced. (See next page)


iptables Process Flow
Incoming Packet
raw PREROUTING
Connection (state) Tracking
mangle PREROUTING
nat PREROUTING
For this host?
N
Y
mangle INPUT
filter INPUT
Local Processing
mangle FORWARD
filter FORWARD
mangle POSTROUTING
nat POSTROUTING
Outgoing Packet
mangle OUTPUT
nat OUTPUT
Routing Decision
filter OUTPUT
Locally-generated Packet
Routing Decision
raw OUTPUT
Connection (state) Tracking

Tables and Chains



Rules

- Table: which table will this rule be added to?
- Chain: which chain will this rule be added to?
- Criteria: properties of a packet or a connection to be matched against.
- Target: the action to be taken if a packet matches the criteria of this rule.

An Example

Rules for a recursive DNS server to allow DNS queries and responses to pass through

```
$ sudo iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
$ sudo iptables -A INPUT  -p udp --sport 53 -j ACCEPT
$ sudo iptables -A OUTPUT -p udp --sport 53 -j ACCEPT
$ sudo iptables -A INPUT  -p udp --dport 53 -j ACCEPT
```

A More Realistic Example for a R-DNS server

```
# Allow SSH and HTTP for Admin
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT  -p tdp --dport 80 -j ACCEPT
sudo iptables -A OUTPUT -p tcp -j ACCEPT
# Allow loopback for inter-process communication
iptables -I INPUT 1 -i lo -j ACCEPT
# Allow DNS
sudo iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
sudo iptables -A INPUT  -p udp --sport 53 -j ACCEPT
sudo iptables -A OUTPUT -p udp --sport 53 -j ACCEPT
sudo iptables -A INPUT  -p udp --dport 53 -j ACCEPT
# Set default filter policy for DROP
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Some More Examples

TBA: 25 Most Frequently Used Linux IPTables Rules Examples

Allow all incoming SSH

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Using SSH Tunnel to Evade Firewalls

This is not necessary a **malicious** practice since it can also facilitate secured network access.

- Local Port Forwarding
- Remote Port Forwarding / Reverse SSH Tunneling
- Dynamic Port Forwarding

SSH Local Port Forwarding.

Evade a firewall that stops incoming telent traffic.

```
$ssh -L 8000:work:23 seed@apollo
$telnet localhost 8000
```

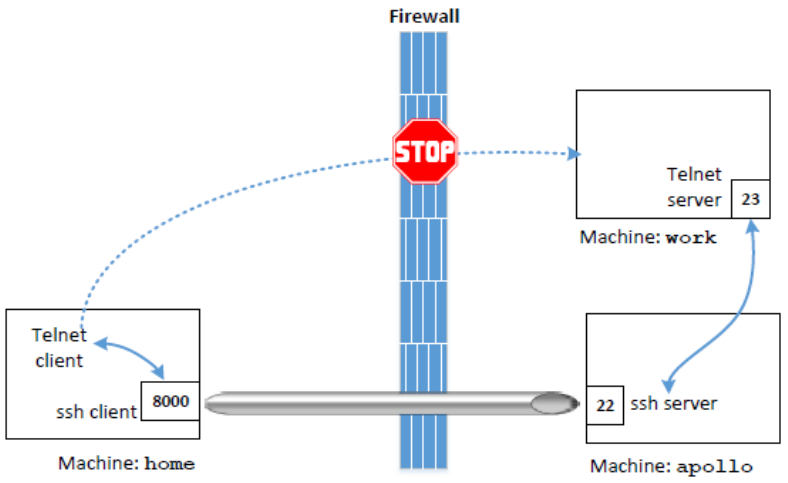


Figure 1: SSH Tunnel Example

SSH Local Port Forwarding.

Evade a firewall that stops outgoing traffic to `www.facebook.com`.

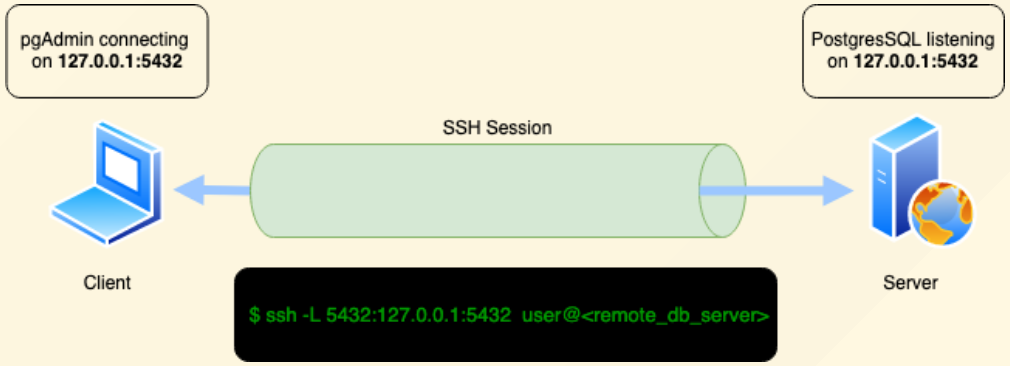
```
$ssh -L 8000:www.facebook.com:80 seed@home
```

Then type `localhost:8000` in your browser to visit `www.facebook.com` via `seed@home`.

SSH Local Port Forwarding.

A very typical, benign, usage of SSH tunnel.

```
bash $ ssh -L 5432:127.0.0.1:5432 user@<remote_db_server>
```



Reverse SSH Tunneling

- The firewall blocks all incoming traffic from outside to an internal web server,say `web-server:80`.
- The firewall blocks all incoming SSH traffic from outside.
- The firewall allow outgoing SSH traffic.
- You have a host, say `appolo`, in internal network.
- You run an SSH server outside, say `seed@home`.

Objective: you want to get access to `web-server:80` from outside.

Reverse SSH Tunneling

```
apollo$ ssh -R 8000:web-server:80 seed@home
```

Now you can send HTTP request to port 8000 on machine `home` from its browser, and the SSH tunnel will forward HTTP requests to the SSH client, `appolo`, which will further forwards the request to the port 80 of the `web-server`.

De-Militarized Zone (DMZ)

A DMZ uses firewalls to expose an organization's external-facing services to an untrusted network such as the Internet.

- An external network node can access only what is exposed in the DMZ.
- The rest of the organization's network is firewalled against the external network.
- The rest of the organization's network can access both DMZ and the Internet.

De-Militarized Zone (DMZ)