Network Address Translation (NAT)

NAT is a process which allows machines in a LAN to share a *single* public IP address.
Every device in the LAN has a *private* IP address that can only be accessed within the LAN. Private IP addresses are of the form 192.168.x.x, 172.16.x.x to 172.31.x.x, etc.

The public IP address is associated with a NAT router, and is the point at which the Internet can access the LAN.

The NAT router is then responsible for managing both inbound and outbound Internet traffic.
How the NAT router works

It maintains a *lookup table* to keep track of the packets that are leaving and entering the LAN.

1. Outbound TCP/UDP Traffic: Suppose an internal host is sending a packet out.
   - The NAT opens a new public source port which will be used for this particular traffic.
   - It creates a new entry for the lookup table.
     - (private source IP, private source port, destination IP, public source port)
   - It then rewrites the source's IP address and port with that of the public IP address and new public source port. It also adjusts any checksums to reflect the changes.
   - The packet is then forwarded to its destination.
How the NAT router works

It maintains a *lookup table* to keep track of the packets that are leaving and entering the LAN.

**Outbound TCP/UDP Traffic:** Suppose an internal host is sending a packet out.

- The NAT opens a **new public source port** which will be used for this particular traffic.
- It creates a new entry for the lookup table.
  (private source IP, private source port, destination IP, public source port)
- It then rewrites the source’s IP address and port with that of the public IP address and new public source port. It also adjusts any checksums to reflect the changes.
- The packet is then forwarded to its destination.
Inbound TCP/UDP Traffic: Suppose the NAT router receives a packet.

- Using the lookup table, it checks if the packet’s source and destination info matches any of its entries.
- If so, the NAT router rewrites the IP headers of the inbound packet according to the entry found in the lookup table so that the right host receives the packet.
• An external machine cannot initiate a connection with a machine in the LAN.
  (–) Violates the ideal of “end-to-end connectivity”.
  (+) On the other hand, it provides some security. In some ways, a NAT router is like a firewall.

• Helps alleviate the exhaustion of IPv4 address space.
  (–) “We should be moving to IPv6! Stop using NAT!!!”
A **firewall** is a mechanism for filtering incoming and outgoing traffic to a machine or a network based on a set of rules called **firewall policies**.

**Example:**
A company’s firewall policies: do not use FTP; do not access websites x, y, z, etc.

The “**Great Firewall of China**”: blocks websites, terminates TCP packets when a certain number of sensitive keywords are used, etc.
Packets flowing through a firewall can have one of three outcomes:

- **Accepted**: permitted through the firewall
- **Dropped**: not allowed through with no indication of failure
- **Rejected**: not allowed through, accompanied by an attempt to inform the source that the packet was rejected
Packets flowing through a firewall can have one of three outcomes:

- Accepted: permitted through the firewall
- Dropped: not allowed through with no indication of failure
- Rejected: not allowed through, accompanied by an attempt to inform the source that the packet was rejected

Rules are based on:

- Protocols used: TCP or UDP, etc.
- the source and destination IP addresses
- the source and destination ports
- the application-level payload of the packet (e.g., whether it contains a virus).
Two fundamental approaches to creating firewall policies:

1. blacklist (or default-allow) rulesets
   - rules define which packets cannot enter
   - flexible, not as disruptive
   - naive: hard to describe malicious traffic
Two fundamental approaches to creating firewall policies:

1. blacklist (or default-allow) rulesets
   - rules define *which packets cannot enter*
   - flexible, not as disruptive
   - naive: hard to describe malicious traffic

2. whitelist (or default deny) rulesets
   - rules define *which packets can enter*
   - provides the greatest possible caution
1. **Stateless Firewalls**

   - Process packets in isolation; simply inspect packets and apply rules.
   - No memory dedicated to determining if a packet is part of an existing connection.
   - Lacks flexibility so it must choose between limited functionality or lax security.
2. Stateful Firewalls

- Like NAT devices, they maintain tables containing info on each active connection.
  TCP session: set-up by the initial handshake process.
  UDP session: “started” when a UDP packet is allowed through the firewall. All subsequent UDP transmissions between the same IP’s and ports are allowed until a specified timeout is reached.
- Allows for more effective policies – e.g. drop connection if number of curse words exceed $x$. 

Countries that do deep packet inspection: US, China, Iran, Russia, etc.
2. Stateful Firewalls

- Like NAT devices, they maintain tables containing info on each active connection.
  - TCP session: set-up by the initial handshake process.
  - UDP session: “started” when a UDP packet is allowed through the firewall. All subsequent UDP transmissions between the same IP’s and ports are allowed until a specified timeout is reached.
- Allows for more effective policies – e.g. drop connection if number of curse words exceed $x$.

**Application-layer Firewalls:** allows for *deep packet inspection*, an examination of the data in an IP packet.

Countries that do deep packet inspection: US, China, Iran, Russia, etc.
A **tunneling** protocol automatically encrypts the communication between a server and a client.

Prevents *eavesdropping*!

**Figure 6.14:** Tunneling protocols provide end-to-end encryption of TCP/IP communication between a client and a server.
The original **SSH** protocol was designed by the Finnish researcher Tatu Ylónen in 1995 to replace telnet, rlogin and rsh. It was prompted by a password sniffing attack at the Helsinki University of Technology.

- Makes use of symmetric and public key encryption and key exchange algorithms.
- Is used for a variety of tasks including secure remote administration, and secure file transfers.
How it works:

1. The client connects to the server via a TCP session.
2. The client and server exchange information on administrative details – e.g. supported encryption methods, their protocol version, etc.
3. [For Encryption] The client and server initiate a secret-key exchange to establish a shared secret session key which will be used for encrypting all of their communication. The most common block ciphers used are AES and 3DES.
4. **[Authentication Step]** The server sends the client a list of acceptable forms of authentication, which the client will try in sequence – e.g., passwords or public-key authentication methods.

For the public-key authentication method:

a. The client sends the server its public key.

b. The server then checks if this key is stored in its list of authorized keys. If so, the server encrypts a challenge using the clients public key and sends it to the client.

c. The client decrypts the challenge with its private key and responds to the server, proving its identity.

5. Once authentication has been successfully completed, the server lets the client access appropriate resources, such as a command prompt.
**IPsec**

**IPsec** defines a set of protocols to provide confidentiality and authenticity for IP packets.

- Created in conjunction with IPv6, but also designed to be backwards compatible with IPv4.
- Operates at the network layer. This means that the standard IP packet has to be modified.
- However, IPsec is transparent to applications – they do not have to change anything!
In IPsec, each protocol can operate in one of two modes, transport mode or tunnel mode.

- **Transport mode**: additional IPsec header information is inserted before the data of the original packet, and only the payload of the packet is encrypted or authenticated.

- **Tunnel mode**: a new packet is constructed with IPsec header information, and the entire original packet, including its header, is encapsulated as the payload of the new packet.
The Authentication Header (AH) Protocol

The **AH protocol** is used to authenticate the origin and guarantee the data integrity of IPsec packets.

**Figure 6.15:** The authentication header.
The authentication data field contains the integrity check value (ICV). The ICV is the MAC value of the entire packet including the IPsec header with the exception of fields that may change during routing and the ICV value itself.

- guarantees data integrity by using a cryptographic hash function
- guarantees authenticity because the MAC requires the use of a secret key.
- cannot modify IP source address because this would modify the ICV; hence, cannot be used with NAT!
The Encapsulating Security Payload (ESP) Protocol

The ESP protocol is used to provide confidentiality to IPsec packets.

**Figure 6.16: The ESP header.**
ESP uses a specified block cipher to encrypt either the entire original IP packet or just its payload.

ESP also provides optional authentication (which can be found in its trailer), which is slightly weaker than that of AH.
Virtual Private Networks (VPNs)

VPN is a technology that allows private networks to be safely extended over long physical distances by making use of a public network, such as the Internet, as a means of transport.
There are two primary types of VPNs:

- **remote access VPN**
  - connects an individual computer to an organization’s network
  - organization sets up a VPN endpoint (also called network access server)
  - individual installs a VPN software on their computer which handles negotiations with the NAS and facilitates communication.

- **site-to-site VPN**
  - connects two or more networks
  - before, leased lines were used!
  - each network must have their own VPN endpoint which is used to communicate with others.
VPNs are NOT standardized. However, most use a limited set of protocols to transfer data to ensure data confidentiality, integrity, and authentication.

Protocols used include IPsec, SSH, etc.
A remote client with public IP address 1.2.3.4 wants to connect to a server inside a company network.

The server has internal address 192.168.1.10 and is not reachable publicly.

To reach this server, the client needs to go through a VPN server/firewall device with public IP address 5.6.7.8 and an internal address of 192.168.1.1.

All data between the client and the server will need to be kept confidential, hence a secure VPN is used.
The steps

1. VPN Client establishes a connection with a VPN Server via an external network interface.

2. VPN Server assigns an IP address to the VPN client from a local virtual subnet say 192.168.1.50.

3. VPN Client prepares a packet for the server 192.168.1.10.

4. VPN Client encrypts and hides an original packet inside the outer public packet. The inner encrypted packet has source address 192.168.1.50 and destination address 192.168.1.10. The outer packet has source address 1.2.3.4 and destination address 5.6.7.8.

5. The packet is dispatched by the VPN client via public network to the VPN Server.
6. The network packet acquired from the public network is decrypted and decapsulated by the VPN server. This way VPN server obtains a packet for the private network.

7. VPN Server handles the newly acquired packet as if it was sent locally. Packet is delivered to the server with destination IP address 192.168.1.10.

8. After some time, server with IP address 192.168.1.10 creates a network packet with destination IP 192.168.1.50.

9. VPN Server receives a reply packet. According to the VPN Server’s routing table, this packet links up with the Virtual Private Network.
10. VPN server encrypts and hides an original packet inside the outer public packet. The inner encrypted packet has source address 192.168.1.10 and destination address 192.168.1.50. The outer VPN packet has source address 5.6.7.8 and destination address 1.2.3.4.

11. The packet is dispatched by the VPN server via public network to the VPN client.

12. The network packet acquired from the public network is decrypted and decapsulated by the VPN Client. This way VPN client obtains a packet from the private network.