Network Address Translation (NAT)

8.10.2007

Adj. Prof. Sasu Tarkoma

Contents

What is NAT

- Expand IP address space by deploying private address and translating them into publicly registered addresses
- Private address space (RFC 1918)
  - 10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
  - 172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
  - 192.168.0.0 - 192.168.255.255 (192.168.0.0/16)
- First described in RFC 1631
- Technique of rewriting IP addresses in headers and application data streams according to a defined policy
- Based on traffic source and/or destination IP address

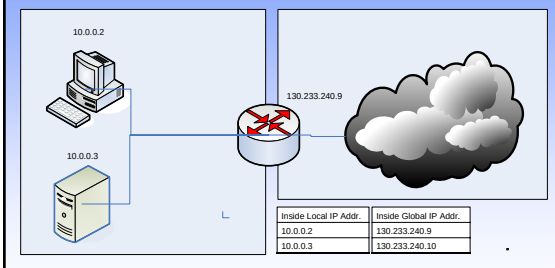
NATs and Firewalls

- Firewalls
  - Security main concern
  - Demilitarized zone
  - Increasingly complex rules (what is filtered, how)
- NATs
  - Lightweight security devices
    - Topology hiding and firewalling
  - Increasing number in deployment
    - Solves some of the address space problems of IPv4 (Port Translation, NAPT)
  - IPv6 solves the addressing problem so NATs are not needed for this

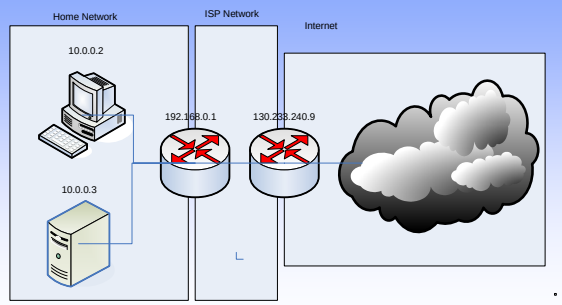
Port Ranges

- Well-known
  - 1-1023
- Registered
  - 1024-49151
- Dynamic/Private
  - 49152-65535
- Port overloading must not be used

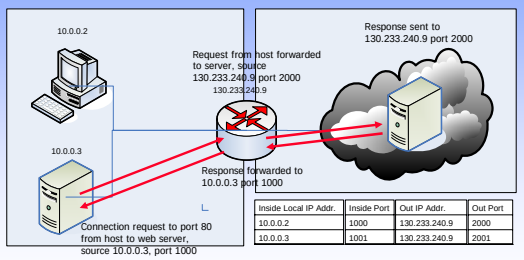
NAT Example



Two NATs



Web Access Example



Inside Local IP Addr.	Inside Port	Out IP Addr.	Out Port
10.0.0.2	1000	130.233.240.9	2000
10.0.0.3	1001	130.233.240.9	2001

NAT traffic supported

Traffic Types/Applications Supported	Traffic Types/Applications not Supported
Any TCP/UDP Traffic that Does Not Carry Source and/or Destination IP Addresses in the Application Data Stream	IP Multicast
HTTP	Routing Table Updates
TFTP	DNS Zone Transfers
Telnet	BOOTP
archie	Talk, Ntalk
finger	H.323
NTP	VDOLive
NFS	NetShow
rlogin, rsh, rcp	VXtreme
Although the Following Traffic Types Carry IP Addresses in the Application Data Stream, they are Supported by Cisco IOS NAT:	SNMP
ICMP	
SMTP	
FTP (Including PORT and PASV Commands)	
NetBIOS over TCP/IP	
Progressive Networks?RealAudio	
White Pines CuSeeMe	
DNS "A" and "PTR" Queries	
Xing Technologies StreamWorks	

Timeouts

- It is crucial how mappings are removed
  - Done using timers
- Mapping refresh
  - Outbound refresh: internal to external
  - Inbound refresh: external to internal
    - May be used, subject to attacks
- A NAT UDP mapping must not expire in less than 2 minutes
- Established connections
  - Must no be less than 2 hours and 4 minutes
  - TCP sends keep alives
- NATs can have application-specific timers

NAT Features

- NAT provides transparent and bi-directional connectivity between networks having arbitrary addressing schemes
- NAT eliminates costs associated with host renumbering
- NAT conserves IP addresses
- NAT eases IP address management
- Load Balancing
- NAT enhances network privacy
- Address migration through translation
- IP masquerading
- Load balancing

NAT Concerns

- Performance
  - IP address modification, NAT boxes need to recalculate IP header checksum
  - Port number modification requires TCP checksum recalculation
- Fragmentation
  - Fragments should have the same destination
- End-to-end connectivity
  - NAT destroys universal end-to-end reachability
  - NATted hosts are often unreachable

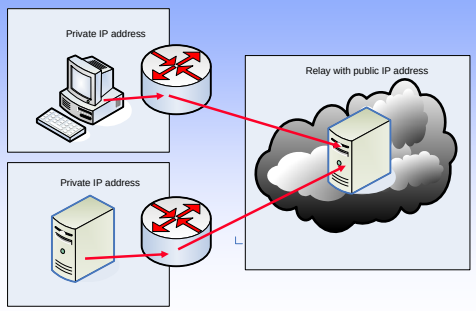
NAT Concerns

- Applications with IP-address content
  - Need AGL (Application Level Gateway)
  - Typically applications that rely on IP addresses in payload do not work across a private-public network boundary
  - Some NATs can translate IP addresses in payload
- NAT device can be a target for attacks
- NAT behaviour is not deterministic
- NATs attempt to be transparent
  - Challenges for network troubleshooting

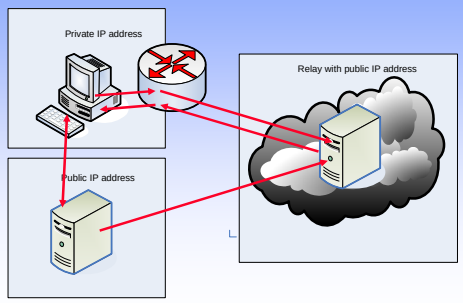
NAT Traversal

- Challenge: how to allow two natted hosts communicate?
- Straighforward solution: use a relay with a public address that is not natted
  - Connection reversal possible if a node has a public address
    - Relay is a rendezvous point
- More complicated solutions
  - Detect presence of NATs
  - Hole punching

Relaying



Connection reversal



NAT Policies I/II

- RFC3489
- Full cone NAT
  - All requests from the same internal IP address and port are mapped to the same public IP address and port.
  - Once a mapping is created, all incoming traffic to the public address is routed to the internal host without checking the address of the remote host.
- Restricted cone NAT
  - Unlike a full cone NAT, a remote host (with IP address X) can send a packet to the internal host only if the internal host had previously sent a packet to IP address X.

NAT Policies II

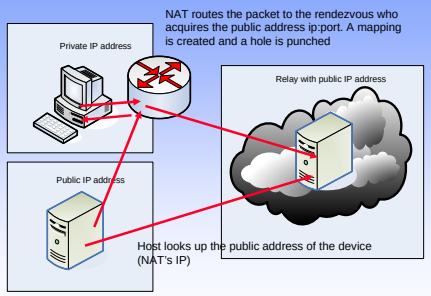
- Port restricted cone NAT
  - A port restricted cone NAT is like a restricted cone NAT, but the restriction includes port numbers.
  - An external host can send a packet, with source IP address X and source port P, to the internal host only if the internal host had previously sent a packet to IP address X and port P.
- Symmetric NAT
  - A symmetric NAT is a NAT where all requests from the same internal IP address and port to a specific destination IP address and port are mapped to the same external source IP address and port. If the same internal host sends a packet with the same source address and port to a different destination, a different mapping is used.
  - **Only the external host that receives a packet can send a UDP packet back to the internal host**

Hole Punching

Hole Punching

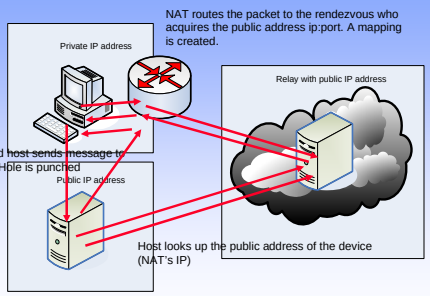
- Hole punching is a technique to allow traffic from/to a host behind a firewall/NAT without the collaboration of the NAT itself
- The simplest way is to use UDP packets

Hole Punching with Full Cone



No restrictions on IP traffic arriving at the NAT

Hole Punching with Restricted Cone

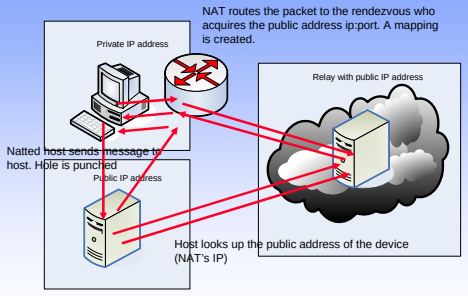


Restricts traffic based on public IP address (not on port)

(X,y) sends to (A,z) through (N,q)

(A,w) can send back to (N,q)

Hole Punching with Port Restricted Cone



Similar as restricted cone, but source port of remote generated packed must be the same addressed by host A in the first packet.

Symmetric NATs

- Punching a hole in symmetric NAT is impossible
  - NAT assigns new mappings for different destinations
  - Random port numbers
  - Only recipient can send packet back to internal host
- Rendezvous is not useful here because NAT will assign new public address to packets for other hosts than the rendezvous
- The only way to traverse this NAT is by Connection Reversal or Relaying.

Additional Note

- A symmetric NAT does not maintain a consistent port binding.It creates a new mapping/binding for each new session. This means that hole punching does not work, because there is no consistent opening for incoming packets.

TURN, STUN, ICE

TURN

- IETF MIDCOM draft ”Traversal Using Relay NAT (TURN)”
- TURN is a protocol for UDP/TCP relaying behind a NAT
- Unlike STUN there is no hole punching and data are bounced to a public server called the TURN server
- TURN is the last resource. For instance behind a symmetric NAT
- It introduces a relay
  - Located in customers DMZ or Service Provider network
  - Single point of failure
  - Requires a high performance server

TURN Elements

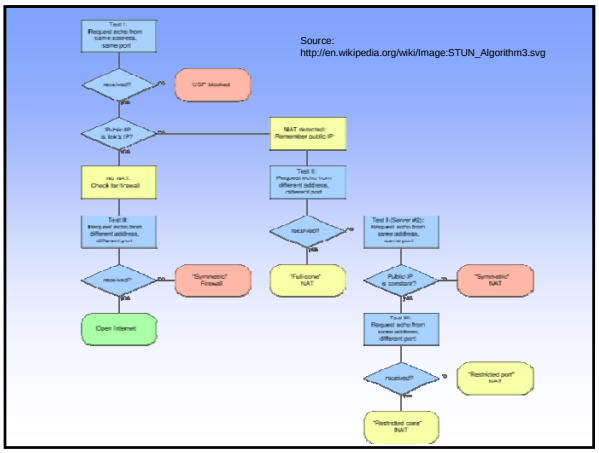
- A TURN client is an entity that generates TURN requests
- A TURN Server is an entity that receives TURN requests, and sends TURN responses.
- The server is a data relay, receiving data on the address it provides to clients, and forwarding them to the clients

STUN

- IETF RFC 3489 ”STUN – Simple Traversal of User Datagram Protocol (UDP) Trough Network Address Translators (NATs)”
- A client-server protocol to discover the presence and types of NAT and firewalls between them and the public Internet
- STUN allows applications to determine the public IP addresses allocated to them by the NAT
- Defines the operations and the message format needed to understand the type of NAT
- The STUN server is contacted on UDP port 3478
- The server will hint clients to perform tests on alternate IP and port number too (STUN servers have two IP addresses)

How STUN works

- The client sends a request to the server
- The server returns a response which contains the source IP of the packet received from the client (inside the payload) i.e. the mapped IP address
- The client compares its IP address with the mapped IP address returned by the server
- If they are different then the client is behind a NAT
- The client can set some flags in the message which tell the server to send a packet from another IP/port or to another IP/port (both the client and the server have 2 IP addresses)



Source:
http://en.wikipedia.org/wiki/Image:STUN_Algorithm3.svg

STUN Limitations

- Does not work with symmetric NATs used by most corporate environments
- Does not work if both clients are behind the same NAT
- Additional deployment of a STUN server placed in the public space

Additional Note

- Once a client has discovered its external addresses, it can relate it to its peers. If the NATs are full cone then either side can initiate communication. If they are restricted cone or restricted port cone both sides must start transmitting together.
- The trick is using STUN to discover the presence of NAT, and to learn and use the bindings they allocate.

Revised STUN

- Revised specifications
  - Classical STUN not deployable
  - "Session Traversal Utilities for NAT"
  - STUN is not a complete solution, rather it is a tool
    - ICE
    - SIP Outbound, ..
  - Binding discovery, NAT keep-alives, Short-term password, Relay (previously TURN)
- Can run on UDP, TCP, TLS
- http://www.ietf.org/internet-drafts/draft-ietf-behave-rfc3489bis-10.txt

STUN Relay

- Previously TURN
- STUN server located using DNS SRV records
- Allocate request/response
  - Allocate an external address at the relay
- Send indication
  - Data to remote endpoint using relay
- Data indication
  - Data received from remote endpoints using relay
- Connect request and response
  - Request relay to establish TCP connection with remote endpoint
- Connection status indication
  - Relay informs endpoint about status of TCP connection: LISTEN, ESTABLISH, CLOSED

ICE

- IETF MMUSIC draft ”Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT)”
- Allows peers to discover NAT types and client capabilities
- Provide alternatives for establishing connectivity, namely STUN and TURN
- Works with all types of NATs, P2P NAT traversal
- Designed for SIP and VoIP. Can be applied to any session-oriented protocol
- The detailed operation of ICE can be broken into six steps: gathering, prioritizing, encoding, offering and answering, checking, and completing.

ICE Step 1

- Gathering
  - Caller gathers IP addresses and ports
  - Each is a potential candidate for communications
  - A candidate is gathered from each interface
  - The agent contacts STUN server from each host interface
  - Result is a set of server-reflexive candidates
    - IP addresses that route to the outermost NAT between the agent and the STUN server
  - Relayed candidates from TURN servers
    - IP addresses and ports on the relay servers

ICE Step 2

- Prioritizing
  - Each candidate is assigned a priority value
- Prioritizing addresses
  - Priority = 2^{24} (type preference) + 2^{8} (local preference) + 2 (256 – component ID)

ICE Step 3

- Encoding
  - Body of a SIP request contains SDP message
  - IP addresses and port numbers
  - ICE extends SDP by several new SDP attributes
    - Candidate attribute, used to transfer candidates (IP address, port, priority)
    - Credential information for securing STUN messaging

ICE Step 4

- Offering and Answering
  - SIP INVITE + SDP is sent to the called party
  - The called party performs the same gathering, prioritizing, and encoding that the caller performed
  - A provisional SIP response with SDP message including the candidates

ICE Step 5

- Checking
  - The caller and called party have exchanged SDP messages
  - ICE performs the work of selecting the communication parameters
  - Each agent computes a priority for the candidate pair by combining the priority of each candidate
- Verification
  - ICE uses a STUN transaction from each agent towards the other to find working candidate pairs
    - Connectivity check
  - Checks are performed sequantially (n squared!)
  - Ordered by priority, check every 20 ms
    - When receives a STUN request, generate a request in the other direction (triggered check)

ICE Step 6

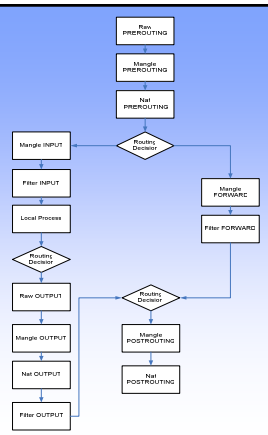
- Completing
  - Once a check is complete, the agent knows that it has found a working pair
  - A final check is generated towards the other agent
  - After this final transaction is sent, the SIP phone will ring (called agent)
  - Answering will generate a SIP 200 OK message
- ICE increases connection setup delays

NATs and SIP

- Client will generate SIP INVITE and 200 OK responses with private addresses
  - In the SDP as the target for receipt of media
    - Solved by ICE (difficult)
  - In the Contact of a REGISTER
    - Solved by SIP Outbound
  - In the Via of a request
    - Solved by rport (RFC 3581)
- Recipient will not be able to send messages to the private address
  - Incoming calls do not work
  - Media will be discarded
  - Responses are not received

IPTABLES

- IPTABLES is a tool for firewall and NAT functionality
- Uses several tables and each table can have several chains
- Each table/chain can have rules that determine how packets are handled
- PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING
- NAT is done with PREROUTING, POSTROUTING
- Firewall is done with INPUT, FORWARD, OUTPUT
- Tables: nat, filter, mangle, raw



IETF BEHAVE WG

- IETF BEHAVE Working Group is working on classification of NAT behaviours
  - Behavior Engineering for Hindrance Avoidance
- Recommendations for NAT vendors
  - Towards deterministic operation
- http://www.ietf.org/html.charters/behave-charter.html

Summary

- NATs are popular on the Internet
  - Private address spaces
  - End-to-end reachability problems
- Four main types of NATs
  - Hole punching and relays basic mechanisms
- A number of solutions for NAT traversal
  - STUN
  - TURN
  - ICE
  - SIP Client-initiated Outbound Connections

Additional Notes