The reality of Network Address Translators

by

Harald Welte <laforge@netfilter.org>

Contents

□RFC3489: STUN

□RFC3714: IAB problem statement / congestion control

□RFC3448: TFRC, TFRC-PS

□DCCP

□NSIS: GIMPS / NAT NSLP

□BEHAVE

NAT Basics

□ Network Address Translation is an old technique

□ Widely used throughout the net as a way to cope with address shortage

□ More and more popular with to DSL and cable modem routers

□ Unfortunately not standardized at all

□ NAT itself is not a security technology !!

NAT Basics

- What does NAT do?
  - Rewrite addresses of packets as they pass a particular forwarding machine

- What can be translated?
  - Layer 3 (IP) addresses
  - Layer 4 (TCP/UDP/SCTP/...) specific addresses
  - Layer 5+ (e.g. FTP PORT statements)

- Where can it be translated?
  - Traditionally, at a router
  - But also possible on a bridge

NAT Configurations

- Source NAT
  - source address of the first packet of a particular connection is changed
- Masquerading
  - special case of Source NAT, most common implementation
- Destination NAT
  - destination address of of the first packet of a particular connection is changed
  - sometimes referred to as ’port mapping’ or ’port redirection’
- Bi-NAT
  - 1:1 translation of whole address ranges or networks

Why is NAT a nightmare

- NAT might have been a solution 8 years ago
- However,
  - it is very much designed for the traditional client/server paradigm
  - the Internet sees more advanced applications such as
    - peer-to-peer networks
    - Voice over IP
    - Multimedia streams
  - protocols are getting increasingly complex
    - multiple layer 4 connections comprising one logical connection
    - embedding layer 3/4 addresses in payload leads to ALG requirement
    - direct ’client-to-client’ transmission of media streams not possible due to deployment of NAT.

NAT Basics

- But well, even eight years ago....
- NATing a FTP connection is a real PITA. Why?
  - First you change the source ip/port of the control connection
  - Then your ftp client sends a PORT command (in ASCII!!!)
    - PORT 123,123,123,123,1,0
  - Then your ftp nat ALG needs to change that to
    - PORT 1,1,1,1,10,10
  - Thus, the resulting string is shorter!
    - therefore you need to mangle every sequence number of each successive packet
    - now think of multiple port commands being issued within a single TCP window and retransmissions
    - if that is not enough, think of SACK
  - Summary
    - It is ugly as hell
    - Difficult to impossible to get right in all cases

Why is NAT a nightmare

- Todays NAT’s horribly violate the network layering model
  - a NAT (although it operats on a rotuer or bridge) requires knowledge of the application protocols
  - support for every new protocol needs to be added to all NAT’s
- Also, you loose the ability to encrypt the payload
  - SIP can PGP-encrypt SDP.
  - However, port numbers are inside SDP
  - Therefore, if you use crypto, it just can’t work

Types of NAT (STUN RFC3489)

- Full Cone
  - all requests from the same internal IP and port are mapped to the same external IP address and port
  - any external host can send a packet to the internal host by sending a packet to the mapped address
- Restricted Cone
  - all requests from the same internal IP and port are mapped to the same external IP address and port.
  - an external host can send a packet to the internal host only if the internal host had previously ent a packet to that particular external host

Types of NAT (STUN RFC3489)

- Port Restricted Cone
  - like restricted cone, but includes port numbers
  - an external host can send a packet with source IP X and port P to the internal host only of the internal host had perviously sent a packet to IP address X and port P
- Symmetric
  - all requests from same internal IP address and port to a specifica destination IP and port are mapped to the same external IP and port.
  - if the same host sends a packet with the same source address and port, but to a different estination, a different mapping is used. Only the external host that receives a packet can send a packet back to the external host

Types of NAT: draft-audet-nat-behave

- Address and port binding
  - External NAT binding is endpoint independent
  - External NAT binding is endpoint address dependent
  - External NAT binding is endpoint address and port dependent
- Port Assignment
  - Port Preservation
  - Port Overloading
- Bind Refresh Scope
  - Per binding
  - Per session
  - Only outgoing or also incoming?

Types of NAT: draft-audet-nat-behave

- Filtering of unsolicited packets
  - External filtering is endpoint independent
  - External filtering is endpoint address dependent
  - External filtering is endpoint address and port dependent
- Hairpinning Behaviour
  - What happens if two endpoints are behind same nat
- Deterministic Properties
  - Chaning over time:
    - Port preservation
    - Port allocation algorithm
    - Address and port binding
    - Filtering
- Multicast Behaviour

The IETF and NAT

- The IETF has long ignored the fact that NAT’s are commonplace
  - Therefore, there’s a lack of standardization in NAT behaviour
  - Furthermore, it is impossible to make a protocol work with all existing NAT’s
  - Protocol designers normally don’t consider NAT when developing new protocols

The IETF and NAT

- SIP was the first IETF protocol that had _serious_ NAT issues
  - Therefore, the SIP working group came up with FCP (Firewall Control Protocol)
  - Later, a new working group ’MIDCOM’ was founded
  - MIDCOM took several years but didn’t really come up with a solution
- Now there are dozens of groups publishing papers, drafts and RFC’s.
- Most of them are targeted at UDP-only operation
- Most of them target consumer side NAT devices

How to solve the NAT problem?

- At a protocol level
  - designing protocols in a way to operate on most/all NAT’s
  - SIP has some extensions for this
  - IPsec also introduced NAT-T to tackle the problem
  - Very difficult because of the number of differnet implementations and lack of standardization

- At a NAT level
  - Making NAT’s interoperate with all different kinds of protocols
  - Support operations like hole-punching for UDP and TCP
  - Problematic because of large existing deployment
-

How to solve the NAT problem?

□ With a specific NAT configuration protocol

- ○ FCP
- ○ MIDCOM
- ○ GIMPS NSIS NAT NSLP
- ○ uPnP

□

□ There is no good solution without standardization

RFC3489: STUN

RFC3489: STUN (Simple Traversal of UDP Through NAT)

- □ Helps endpoints to find out whether they are behind some form of NAT by communication with a host known to have an official IP
- □ Tries to create NAT binding(s) on NAT devices
- □ allows applications to ’open ports’ on the NAT
- □ implemented with lots of apps, including gnomemeeting

RFC3714

- IAB problem statement about media traffic without congestion control
  - danger of congestion collapse with VoIP / streaming media
  - IETF actions to counter this problem
    - upgrade RTP to make packet loss monitoring a MUST
    - TFRC (TCP Friently Rate Control)
    - TFRC-PS (TCP Friendly Rate Control - Packet Size)
    - DCCP (Datagram Congestion Control Protocol)
    - Adaptive Audio Codecs
    - specified drop rate for mimimum sending rate (tables)
- Result:
  - We’ll see new layer four protocols that need NAT, too

NSIS WG

- NSIS (Next Step In Signalling) WG:
  - Signalling Transport protocol for Signalling QoS, NAT, Firewalls
  - GIMPS (Generic Internet Messaging Protocol for Signalling)
    - Builds on top of TCP/UDP/SCTP/DCCP
    - can be combined with TLS and IPsec
    - Has Messages with ’Router Alert’ that are to be processed by Routers/Firewalls/NATs
  - NAT NSIS Signalling Layer Protocol
    - wants to establish a connection between two ends, any number of Firewalls / NAT’s in between
    - draft-aoun-nsis-nslp-natfw-migration-02
    - draft-tschofenig-nsis-natfw-security-problems-00
    - draft-aoun-nsis-nslp-natfw-intrarealm-00.txt
    - draft-martin-nsis-nslp-natfw-sip-00.txt
    - draft-fessi-nsis-natfw-threats-01.txt

BEHAVE

- Behave working group
  - Parts of IETF acknowledge NAT is reality
  - Acknowledges lack of standardization
  - wants to provide vendor guidelines for NAT implementation
  - focus on UDP and TCP unicast
  - will adress multicast NAT, too
  - goal: NAT-BEHAVE BCP RFC
  - second document describing protocol design for BEHAVE-compliant NATs
  - current draft:
    - require outbound-only UDP timer refresh
    - strongly discourages port persistency
    - requires no NAT for IPv6

Thanks

□ Thanks to

- ○ Alan Cox, Alexey Kuznetsov, David Miller, Andi Kleen
  - ▹ for implementing (one of?) the world’s best TCP/IP stacks
- ○ Paul ’Rusty’ Russell
  - ▹ for starting the netfilter/iptables project
  - ▹ for trusting me to maintain it today
- ○ Astaro AG
  - ▹ for sponsoring parts of my netfilter work
- ○ Free Software Foundation
  - ▹ for the GNU Project
  - ▹ for the GNU General Public License

□ The slides of this presentation are available at http://www.gnumonks.org/

□ Further Reading

□ The netfilter homepage http://www.netfilter.org/