NAT

Document revision 2.9 (February 11, 2008, 4:14 GMT)
This document applies to V3.0

Table of Contents

General Information

Summary

Network Address Translation (NAT) is a router facility that replaces source and (or) destination IP addresses of the IP packet as it pass through thhe router. It is most commonly used to enable multiple host on a private network to access the Internet using a single public IP address.

Specifications

Packages required: ***system***
License required: ***level1 (number of rules limited to 1), level3***
Home menu level: ***/ip firewall nat***
Standards and Technologies: ***IP***, ***RFC1631***, ***RFC2663***
Hardware usage: *Increases with the count of rules*

NAT

Description

Network Address Translation is an Internet standard that allows hosts on local area networks to use one set of IP addresses for internal communications and another set of IP addresses for external communications. A LAN that uses NAT is referred as *natted* network. For NAT to function, there should be a NAT gateway in each natted network. The NAT gateway (NAT router) performs IP address rewriting on the way a packet travel from/to LAN.

There are two types of NAT:

- source NAT or srcnat. This type of NAT is performed on packets that are originated from a natted network. A NAT router replaces the private source address of an IP packet with a new public IP address as it travels through the router. A reverse operation is applied to the reply packets travelling in the other direction.
- destination NAT or dstnat. This type of NAT is performed on packets that are destined to the natted network. It is most comonly used to make hosts on a private network to be acceesible from the Internet. A NAT router performing dstnat replaces the destination IP address of an IP packet as it travel through the router towards a private network.

NAT Drawbacks

Hosts behind a NAT-enabled router do not have true end-to-end connectivity. Therefore some Internet protocols might not work in scenarios with NAT. Services that require the initiation of TCP connection from outside the private network or stateless protocols such as UDP, can be disrupted. Moreover, some protocols are inherently incompatible with NAT, a bold example is AH protocol from the IPsec suite.

RouterOS includes a number of so-called NAT helpers, that enable NAT traversal for various protocols.

Redirect and Masquerade

Redirect and masquerade are special forms of destination NAT and source NAT, respectively. Redirect is similar to the regular destination NAT in the same way as masquerade is similar to the source NAT - masquerade is a special form of source NAT without need to specify **to-addresses** - outgoing interface address is used automatically. The same is for redirect - it is a form of destination NAT where **to-addresses** is not used - incoming interface address is used instead. Note that **to-ports** is meaningful for redirect rules - this is the port of the service on the router that will handle these requests (e.g. web proxy).

When packet is dst-natted (no matter - **action=nat** or **action=redirect**), dst address is changed. Information about translation of addresses (including original dst address) is kept in router's internal tables. Transparent web proxy working on router (when web requests get redirected to proxy port on router) can access this information from internal tables and get address of web server from them. If you are dst-natting to some different proxy server, it has no way to find web server's address from IP header (because dst address of IP packet that previously was address of web server has changed to address of proxy server). Starting from HTTP/1.1 there is special header in HTTP request which tells web server address, so proxy server can use it, instead of dst address of IP packet. If there is no such header (older HTTP version on client), proxy server can not determine web server address and therefore can not work.

It means, that it is impossible to correctly transparently redirect HTTP traffic from router to some other transparent-proxy box. Only correct way is to add transparent proxy on the router itself, and configure it so that your "real" proxy is its parent-proxy. In this situation your "real" proxy does not have to be transparent any more, as proxy on router will be transparent and will forward proxy-style requests (according to standard; these requests include all necessary information about web server) to "real" proxy.

Property Description

action (*accept* | *add-dst-to-address-list* | *add-src-to-address-list* | *dst-nat* | *jump* | *log* | *masquerade* | *netmap* | *passthrough* | *redirect* | *return* | *same* | *src-nat*; default: **accept**) - action to undertake if the packet matches the rule

- **accept** - accepts the packet. No action is taken, i.e. the packet is passed through and no more rules are applied to it
- **add-dst-to-address-list** - adds destination address of an IP packet to the address list specified by address-list parameter
- **add-src-to-address-list** - adds source address of an IP packet to the address list specified by address-list parameter
- **dst-nat** - replaces destination address of an IP packet to values specified by to-addresses and to-ports parameters
- **jump** - jump to the chain specified by the value of the jump-target parameter
- **log** - each match with this action will add a message to the system log
- **masquerade** - replaces source address of an IP packet to an automatically determined by the routing facility IP address
- **netmap** - creates a static 1:1 mapping of one set of IP addresses to another one. Often used to distribute public IP addresses to hosts on private networks
- **passthrough** - ignores this rule goes on to the next one
- **redirect** - replaces destination address of an IP packet to one of the router's local addresses
- **return** - passes control back to the chain from where the jump took place
- **same** - gives a particular client the same source/destination IP address from supplied range for each connection. This is most frequently used for services that expect the same client address for multiple connections from the same client
- **src-nat** - replaces source address of an IP packet to values specified by to-addresses and to-ports parameters

address-list (*name*) - specifies the name of the address list to collect IP addresses from rules having action=add-dst-to-address-list or action=add-src-to-address-list actions. These address lists could be later used for packet matching

address-list-timeout (*time*; default: **00:00:00**) - time interval after which the address will be removed from the address list specified by address-list parameter. Used in conjunction with add-dst-to-address-list or add-src-to-address-list actions

- **00:00:00** - leave the address in the address list forever

chain (*dstnat* | *srcnatname*) - specifies the chain to put a particular rule into. As the different traffic is passed through different chains, always be careful in choosing the right chain for a new rule. If the input does not match the name of an already defined chain, a new chain will be created

- **dstnat** - a rule placed in this chain is applied before routing. The rules that replace destination addresses of IP packets should be placed there
- **srcnat** - a rule placed in this chain is applied after routing. The rules that replace the source addresses of IP packets should be placed there

comment (*text*) - a descriptive comment for the rule. A comment can be used to identify rules form scripts

connection-bytes (*integerinteger*) - matches packets only if a given amount of bytes has already been transfered through the particular connection

- **0** - means infinity, exempli gratia: connection-bytes=2000000-0 means that the rule matches if more than 2MB has been transfered through the relevant connection

connection-limit (*integernetmask*) - restrict connection number per address or address block

(matches if the specified number of connection has already been established)

connection-mark (*name*) - matches packets marked via mangle facility with particular connection mark

connection-type (*ftp* | *gre* | *h323* | *irc* | *mms* | *pptp* | *quake3* | *tftp*) - matches packets from related connections based on information from their connection tracking helpers. A relevant connection helper must be enabled under /ip firewall service-port

content (*text*) - the text packets should contain in order to match the rule

dscp (*integer*: 0..63) - DSCP (ex-ToS) IP header field value

dst-address (*IP addressnetmaskIP addressIP address*) - specifies the address range an IP packet is destined to. Note that console converts entered address/netmask value to a valid network address, i.e.:1.1.1.1/24 is converted to 1.1.1.0/24

dst-address-list (*name*) - matches destination address of a packet against user-defined address list

dst-address-type (*unicast* | *local* | *broadcast* | *multicast*) - matches destination address type of the IP packet, one of the:

- **unicast** - IP addresses used for one point to another point transmission. There is only one sender and one receiver in this case
- **local** - matches addresses assigned to router's interfaces
- **broadcast** - the IP packet is sent from one point to all other points in the IP subnetwork
- **multicast** - this type of IP addressing is responsible for transmission from one or more points to a set of other points

dst-limit (*integertimeintegerdst-address* | *dst-port* | *src-addresstime*) - limits the packet per second (pps) rate on a per destination IP or per destination port base. As opposed to the limit match, every destination IP address / destination port has it's own limit. The options are as follows (in order of appearance):

- **count** - maximum average packet rate, measured in packets per second (pps), unless followed by time option
- **time** - specifies the time interval over which the packet rate is measured
- **burst** - number of packets to match in a burst
- **mode** - the classifier(-s) for packet rate limiting
- **expire** - specifies interval after which recorded IP addresses / ports will be deleted

dst-port (*integer*: 0..65535*integer*: 0..65535) - destination port number or range

fragment (yes | no) - whether the packet is a fragment of an IP packet. Starting packet (i.e., first fragment) does not count. Note that is the connection tracking is enabled, there will be no fragments as the system automatically assembles every packet

hotspot (*multiple choice: auth* | *from-client* | *http* | *local-dst* | *to-client*) - matches packets received from clients against various HotSpot conditions. All values can be negated

- **auth** - true, if a packet comes from an authenticted HotSpotclient
- **from-client** - true, if a packet comes from any HotSpot client
- **http** - true, if a HotSpot client sends a packet to the address and port previously detected as his proxy server (Universal Proxy technique) or if the destination port is 80 and transparent proxying is enabled for that particular client
- **local-dst** - true, if a packet has local destination IP address

- **to-client** - true, if a packet is sent to a client

icmp-options (*integerinteger*) - matches ICMP Type:Code fields

in-bridge-port (*name*) - actual interface the packet has entered the router through (if bridged, this property matches the actual bridge port, while in-interface - the bridge itself)

in-interface (*name*) - interface the packet has entered the router through (if the interface is bridged, then the packet will appear to come from the bridge interface itself)

ingress-priority (*integer*: 0..63) - INGRESS (received) priority of the packet, if set (0 otherwise). The priority may be derived from either VLAN or WMM priority

ipv4-options (*any* | *loose-source-routing* | *no-record-route* | *no-router-alert* | *no-source-routing* | *no-timestamp* | *none* | *record-route* | *router-alert* | *strict-source-routing* | *timestamp*) - match ipv4 header options

- **any** - match packet with at least one of the ipv4 options
- **loose-source-routing** - match packets with loose source routing option. This option is used to route the internet datagram based on information supplied by the source
- **no-record-route** - match packets with no record route option. This option is used to route the internet datagram based on information supplied by the source
- **no-router-alert** - match packets with no router alter option
- **no-source-routing** - match packets with no source routing option
- **no-timestamp** - match packets with no timestamp option
- **record-route** - match packets with record route option
- **router-alert** - match packets with router alter option
- **strict-source-routing** - match packets with strict source routing option
- **timestamp** - match packets with timestamp

jump-target (*dstnat* | *srcnatname*) - name of the target chain to jump to, if the action=jump is used

layer7-protocol (*name*) - Layer 7 filter name as set in the /ip firewall layer7-protocol menu. Caution: this matcher needs high computational power

limit (*integertimeinteger*) - restricts packet match rate to a given limit. Usefull to reduce the amount of log messages

- **count** - maximum average packet rate, measured in packets per second (pps), unless followed by time option
- **time** - specifies the time interval over which the packet rate is measured
- **burst** - number of packets to match in a burst

log-prefix (*text*) - all messages written to logs will contain the prefix specified herein. Used in conjunction with action=log

nth (*integerinteger*: 0..15*integer*) - match a particular Nth packet received by the rule. One of 16 available counters can be used to count packets

- **every** - match every every+1th packet. For example, if every=1 then the rule matches every 2nd packet
- **counter** - specifies which counter to use. A counter increments each time the rule containing nth match matches
- **packet** - match on the given packet number. The value by obvious reasons must be between 0 and every. If this option is used for a given counter, then there must be at least every+1 rules

with this option, covering all values between 0 and every inclusively.

out-bridge-port (*name*) - actual interface the packet is leaving the router through (if bridged, this property matches the actual bridge port, while out-interface - the bridge itself)

out-interface (*name*) - interface the packet is leaving the router through (if the interface is bridged, then the packet will appear to leave through the bridge interface itself)

packet-mark (*text*) - matches packets marked via mangle facility with particular packet mark

packet-size (*integer*: 0..65535*integer*: 0..65535) - matches packet of the specified size or size range in bytes

- **min** - specifies lower boundary of the size range or a standalone value
- **max** - specifies upper boundary of the size range

port (*port*) - matches if any (source or destination) port matches the specified list of ports or port ranges (note that the protocol must still be selected, just like for the regular src-port and dst-port matchers)

protocol (*ddp* | *egp* | *encap* | *ggp* | *gre* | *hmp* | *icmp* | *idrp-cmtp* | *igmp* | *ipencap* | *ipip* | *ipsec-ah* | *ipsec-esp* | *iso-tp4* | *ospf* | *pup* | *rdp* | *rspf* | *st* | *tcp* | *udp* | *vmtp* | *xns-idp* | *xtpinteger*) - matches particular IP protocol specified by protocol name or number. You should specify this setting if you want to specify ports

psd (*integertimeintegerinteger*) - attempts to detect TCP and UDP scans. It is advised to assign lower weight to ports with high numbers to reduce the frequency of false positives, such as from passive mode FTP transfers

- **WeightThreshold** - total weight of the latest TCP/UDP packets with different destination ports coming from the same host to be treated as port scan sequence
- **DelayThreshold** - delay for the packets with different destination ports coming from the same host to be treated as possible port scan subsequence
- **LowPortWeight** - weight of the packets with privileged (<=1024) destination port
- **HighPortWeight** - weight of the packet with non-priviliged destination port

random (*integer*) - match packets randomly with given propability

routing-mark (*name*) - matches packets marked by mangle facility with particular routing mark

same-not-by-dst (*yes* | *no*) - specifies whether to account or not to account for destination IP address when selecting a new source IP address for packets matched by rules with action=same

src-address (*IP addressnetmaskIP addressIP address*) - specifies the address range an IP packet is originated from. Note that console converts entered address/netmask value to a valid network address, i.e.:1.1.1.1/24 is converted to 1.1.1.0/24

src-address-list (*name*) - matches source address of a packet against user-defined address list

src-address-type (*unicast* | *local* | *broadcast* | *multicast*) - matches source address type of the IP packet, one of the:

- **unicast** - IP addresses used for one point to another point transmission. There is only one sender and one receiver in this case
- **local** - matches addresses assigned to router's interfaces
- **broadcast** - the IP packet is sent from one point to all other points in the IP subnetwork
- **multicast** - this type of IP addressing is responsible for transmission from one or more points to a set of other points

src-mac-address (*MAC address*) - source MAC address

src-port (*integer*: 0..65535*integer*: 0..65535) - source port number or range

tcp-mss (*integer*: 0..65535) - matches TCP MSS value of an IP packet

time (*timetimesat* | *fri* | *thu* | *wed* | *tue* | *mon* | *sun*) - allows to create filter based on the packets' arrival time and date or, for locally generated packets, departure time and date

to-addresses (*IP addressIP address*; default: **0.0.0.0**) - address or address range to replace original address of an IP packet with

to-ports (*integer*: 0..65535*integer*: 0..65535) - port or port range to replace original port of an IP packet with

NAT Applications

Description

In this section some NAT applications and examples of them are discussed.

Basic NAT configuration

Assume we want to create router that:

- "hides" the private LAN "behind" one address
- provides Public IP to the Local server
- creates 1:1 mapping of network addresses

Example of Source NAT (Masquerading)

If you want to "hide" the private LAN 192.168.0.0/24 "behind" one address 10.5.8.109 given to you by the ISP, you should use the source network address translation (masquerading) feature of the MikroTik router. The masquerading will change the source IP address and port of the packets originated from the network 192.168.0.0/24 to the address 10.5.8.109 of the router when the packet is routed through it.

To use masquerading, a source NAT rule with action 'masquerade' should be added to the firewall configuration:

```
/ip firewall nat add chain=srcnat action=masquerade out-interface=Public
```

All outgoing connections from the network 192.168.0.0/24 will have source address 10.5.8.109 of the router and source port above 1024. No access from the Internet will be possible to the Local addresses. If you want to allow connections to the server on the local network, you should use destination Network Address Translation (NAT).

Example of Destination NAT

If you want to link Public IP 10.5.8.200 address to Local one 192.168.0.109, you should use destination address translation feature of the MikroTik router. Also if you want allow Local server to talk with outside with given Public IP you should use source address translation, too

Add Public IP to Public interface:

```
/ip address add address=10.5.8.200/32 interface=Public
```

Add rule allowing access to the internal server from external networks:

```
/ip firewall nat add chain=dstnat dst-address=10.5.8.200 action=dst-nat \
        to-addresses=192.168.0.109
```

Add rule allowing the internal server to talk to the outer networks having its source address translated to 10.5.8.200:

```
/ip firewall nat add chain=srcnat src-address=192.168.0.109 action=src-nat \
        to-addresses=10.5.8.200
```

Example of 1:1 mapping

If you want to link Public IP subnet 11.11.11.0/24 to local one 2.2.2.0/24, you should use destination address translation and source address translation features with **action=netmap**.

```
/ip firewall nat add chain=dstnat dst-address=11.11.11.1-11.11.11.254 \
        action=netmap to-addresses=2.2.2.1-2.2.2.254

/ip firewall nat add chain=srcnat src-address=2.2.2.1-2.2.2.254 \
        action=netmap to-addresses=11.11.11.1-11.11.11.254
```