Document: draft-cheshire-nat-pmp-00.txt
Internet-Draft
Expires 7th December 2005

Stuart Cheshire
Marc Krochmal
Kiren Sekar
Apple Computer, Inc.
7th June 2005

NAT Port Mapping Protocol (NAT-PMP)

<draft-cheshire-nat-pmp-00.txt>

Status of this Memo

Abstract

This document describes a protocol for automating the process of creating Network Address Translation (NAT) port mappings. Included in the protocol is a method for retrieving the public IP address of a NAT gateway, thus allowing a client to make this public IP address and port number known to peers that may wish to communicate with it. This protocol is implemented in current Apple products including Mac OS X v10.4 Tiger and Bonjour for Windows.

1. Introduction

Network Address Translation (NAT) is a method of sharing one public internet address with a number of devices. This document is focused on what "IP Network Address Translator (NAT) Terminology and Considerations" [RFC 2663] calls "NAPTs" (Network Address/Port Translators). A full description of NAT is beyond the scope of this document. The following brief overview will cover the aspects relevant to this port mapping protocol. For more information on NAT, see "Traditional IP Network Address Translator" [RFC 3022].

NATs have one or more public IP addresses. A private network is set up behind the NAT. Devices behind the NAT are assigned private addresses and the private address of the NAT device is used as the gateway.

When a packet from any device behind the NAT is sent to an address on the public internet, the packet first passes through the NAT box. The NAT box looks at the source port and address. In some cases, a NAT will also keep track of the destination port and address. The NAT then creates a mapping from the private address and private port to a public address and public port if a mapping does not already exist. The NAT box replaces the private address and port number in the packet with the public entries from the mapping and sends the packet on to the next gateway.

When a packet from any address on the internet is received on the NAT's public side, the NAT will look up the destination port (public port) in the list of mappings. If an entry is found, it will contain the private address and port that the packet should be sent to. The NAT gateway will then rewrite the destination address and port with those from the mapping. The packet will then be forwarded to the new destination addresses. If the packet did not match any mapping, the packet will most likely be dropped. Various NATs implement different strategies to handle this. The important thing to note is that if there is no mapping, the NAT does not know which private address the packet should be sent to.

Mappings are usually created automatically as a result of observing outbound traffic. There are a few exceptions. Some NATs may allow manually-created permanent mappings that map a public port to a specific private IP address and port. Such a mapping allows incoming connections to the device with that private address. Some NATs also implement a default mapping where any inbound traffic that does not match a mapping will always be forwarded to a specific private address. Both types of mappings are usually set up manually through some configuration tool.

Without these manually-created inbound port mappings, clients behind the NAT would be unable to receive inbound connections, which represents a loss of connectivity when compared to the original

Internet architecture [ETEAISD]. For those who view this loss of connectivity as a bad thing, NAT-PMP allows clients to operate much more like a host directly connected to the unrestricted public Internet, with an unrestricted public IP address. NAT-PMP allows client hosts to communicate with the NAT gateway to request the creation of inbound mappings on demand. Having created a NAT mapping to allow inbound connections, the client can then record its public IP address and public port number in a public registry (e.g. the world-wide Domain Name System) or otherwise make it accessible to peers that wish to communicate with it.

2. Conventions and Terminology Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC 2119].

3. Protocol and Packet Format

NAT Port Mapping Protocol runs over UDP. Every packet starts with an 8 bit version followed by an 8 bit operation code.

This document specifies version 0 of the protocol. Any NAT-PMP gateway implementing this version of the protocol, receiving a packet with a version number other than 0, MUST return result code 1

(Unsupported Version).

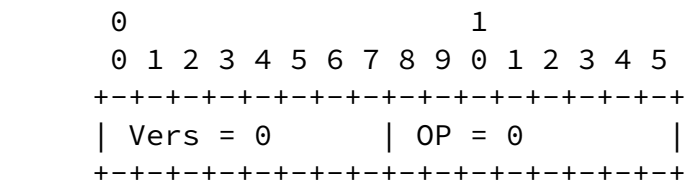
Opcodes between 0 and 127 are client requests. Opcodes from 128 to 255 are server responses. Responses always contain a 16 bit result code in network byte order. A result code of zero indicates success. Responses also contain a 32 bit unsigned integer corresponding to the number of seconds since the NAT gateway was rebooted or since its port mapping state was reset.

This protocol SHOULD only be used when the client determines that its primary source IPv4 address is in the range of private IP addresses defined in RFC 1918. This includes the address ranges 10/8, 172.16/12, and 192.168/16.

Clients always send their Port Mapping Protocol requests to their default gateway, as learned via DHCP [RFC 2131], or similar means. This protocol is designed for small home networks, with a single logical link (subnet) where the client's default gateway is also the NAT translator for that network. For more complicated networks where the NAT translator is some device other than the client's default gateway, this protocol is not appropriate.

3.1 Determining the Public Address

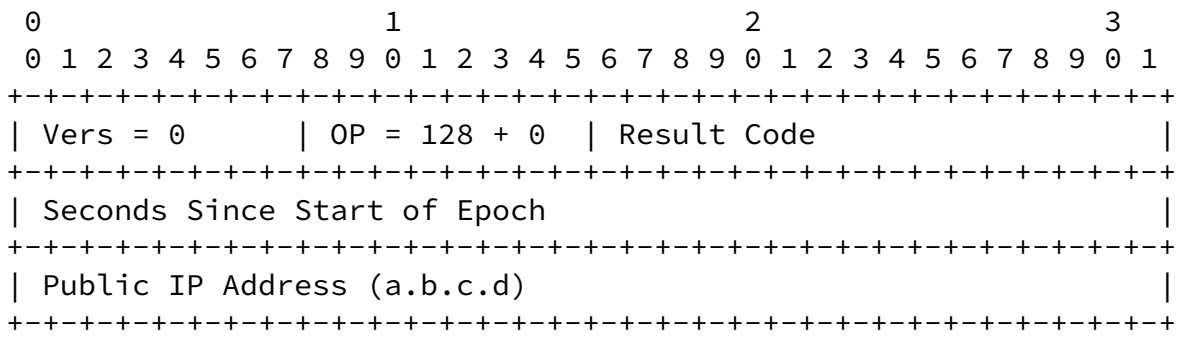
To determine the public address, the client behind the NAT sends the following UDP payload to port 5351 of the configured gateway address:



This packet is sent by clients to determine the public IP address and to determine whether or not the gateway supports this protocol. After sending the request, the client then waits for the NAT gateway to respond. If after 250ms, the gateway doesn't respond (and doesn't generate "ICMP Port Unreachable" messages), the client SHOULD re-issue its request. The client SHOULD repeat this process with the interval between attempts doubling each time. If, after two minutes, the client has not received any response, then it SHOULD conclude

that this gateway does not support NAT Port Mapping Protocol and MAY log an error message indicating this fact.

A compatible NAT gateway MUST generate a response with the following format:



This response indicates that the NAT gateway implements this version of the protocol and returns the public IP address of the NAT gateway. If the result code is non-zero, the value of Public IP Address is undefined (MUST be set to zero on transmission, and MUST be ignored on reception).

The NAT gateway MUST fill in the "Seconds Since Start of Epoch" field with the time elapsed since its port mapping table was initialized on startup or reset for any other reason (see Section 3.7 "Seconds Since Start of Epoch").

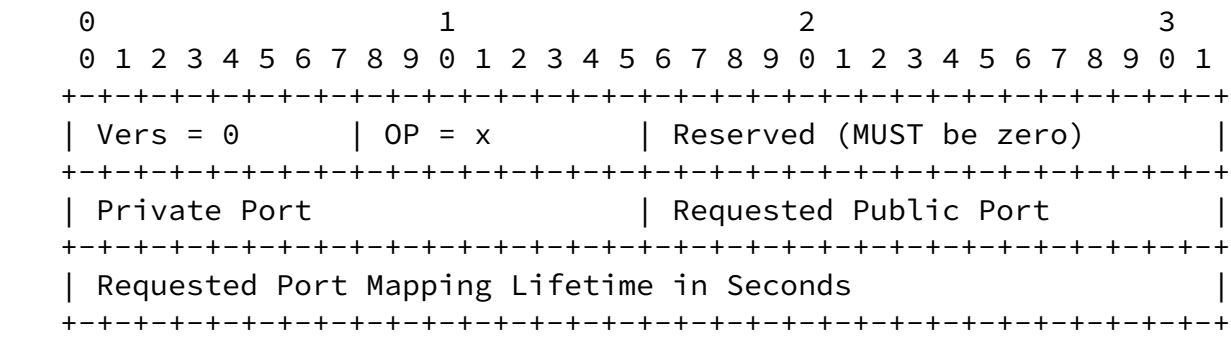
If the client receives an "ICMP Port Unreachable" message from the gateway, then it SHOULD conclude that this gateway does not support NAT Port Mapping Protocol and MAY log an error message.

3.1.1 Announcing Address Changes

When the public IP address of the NAT changes, the NAT gateway MUST send a gratuitous response to the link-local multicast address 224.0.0.1, port 5351 with the packet format above to notify clients of the new public IP address. To compensate for packet loss, the NAT gateway SHOULD multicast 10 address change notifications. The interval between the first two notifications SHOULD be 250ms, and the interval between each subsequent notification SHOULD double.

3.2 Creating a Mapping

To create a mapping, the client sends a UDP packet to port 5351 of the gateway's private IP address with the following format:

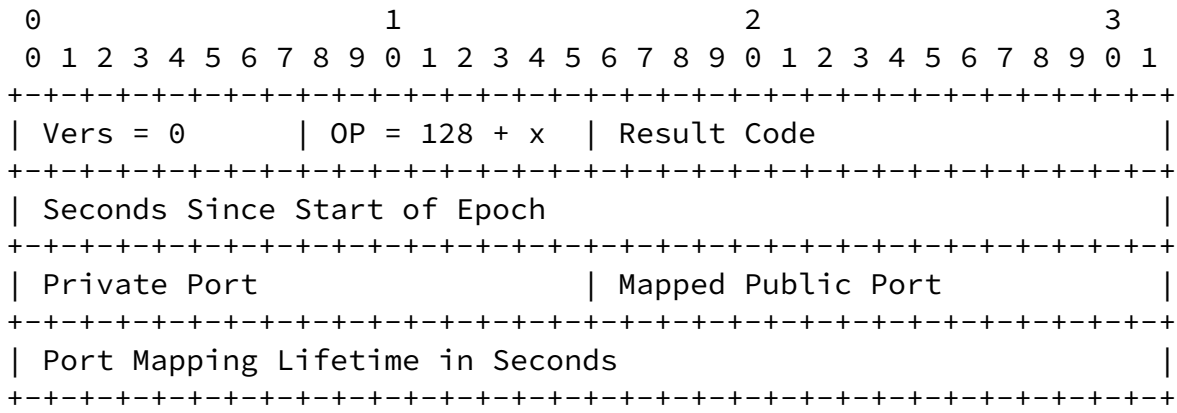


Opcodes supported:
1 - Map UDP
2 - Map TCP

The reserved field MUST be set to zero on transmission and MUST be ignored on reception. The RECOMMENDED Port Mapping Lifetime for this protocol is 3600 seconds.

After sending the port mapping request, the client then waits for the NAT gateway to respond. If after 250ms, the gateway doesn't respond, the client SHOULD re-issue its request. The client SHOULD repeat this process with the interval between attempts doubling each time. If after two minutes, the client has not received any response, it SHOULD log an error message and give up.

The NAT gateway responds with the following packet format:



The 'x' in the OP field MUST match what the client requested. Some NAT gateways are incapable of creating a UDP port mapping without also creating a corresponding TCP port mapping, and vice versa, and these gateways MUST NOT implement NAT Port Mapping Protocol until this deficiency is fixed. A NAT gateway which implements this protocol MUST be able to create TCP-only and UDP-only port mappings. If a NAT gateway silently creates a pair of mappings for a client that only requested one mapping, then it may expose that client to receiving inbound UDP packets or inbound TCP connection requests that it did not ask for and does not want.

While a NAT gateway MUST NOT automatically create mappings for TCP when the client requests UDP, and vice versa, the NAT gateway SHOULD reserve the companion port so the same client can choose to map it in the future. For example, if a client requests to map TCP port 80, as long as the client maintains the lease for that TCP port mapping, another client with a different IP address SHOULD NOT be able to successfully acquire the mapping for UDP port 80.

The client normally requests the public port matching the private port. If that public port is not available, the NAT gateway MUST return a public port that is available or return an error code if no ports are available. If the client has no preference about what public port is assigned, then it should set the requested public port to zero.

The source address of the packet MUST be used for the private address in the mapping. This protocol is not intended to facilitate one device behind a NAT creating mappings for other devices. If there are legacy devices that require inbound mappings, permanent mappings can be created manually by the administrator, just as they are today.

If a mapping already exists for a given private port on a given local client, and that client requests another mapping for the same private port, possibly requesting a different public port, then the mapping request should succeed, returning the already-assigned public port. This is necessary to handle the case where a client requests a mapping with requested public port X, and is granted a mapping with actual public port Y, but the acknowledgement packet gets lost. When the client retransmits its mapping request, it should get back the same positive acknowledgement as was sent the first time.

The NAT gateway SHOULD NOT accept mapping requests destined to the NAT gateway's public IP address or received on its public network interface. Only packets received on the private interface(s) with a destination address matching the private address(es) of the NAT gateway should be allowed.

The NAT gateway MUST fill in the "Seconds Since Start of Epoch" field with the time elapsed since its port mapping table was initialized on startup or reset for any other reason (see Section 3.7 "Seconds Since Start of Epoch").

The Port Mapping Lifetime is an unsigned integer in seconds. The NAT gateway MAY reduce the lifetime from what the client requested. The NAT gateway SHOULD NOT offer a lease lifetime greater than that requested by the client.

The client SHOULD begin trying to renew the mapping halfway to expiry time, like DHCP. The renewal packet should look exactly the same as a request packet, except that the client SHOULD set the requested public port to what the router previously mapped. If the router crashes or is rebooted, this helps the router recover its mapping state.

3.3 Destroying a Mapping

A mapping may be destroyed in a variety of ways. If a client fails to renew a mapping, then when its lifetime expires the mapping MUST be automatically deleted. In the common case where the gateway device is a combined DHCP server and NAT gateway, when a client's DHCP address lease expires, the gateway device MAY automatically delete any mappings belonging to that client. Otherwise a new client being assigned the same IP address could receive unexpected inbound

UDP packets or inbound TCP connection requests that it did not ask for and does not want.

A client MAY also send an explicit packet to request deletion of a mapping that is no longer needed. A client requests explicit deletion of a mapping by sending a message to the NAT gateway requesting the mapping, with the Requested Lifetime in Seconds set to 0. The requested public port MUST be set to zero by the client on sending, and MUST be ignored by the gateway on reception.

When a mapping is destroyed successfully as a result of the client explicitly requesting the deletion, the NAT gateway MUST send a response packet which is formatted as defined in section 3.2. The response MUST contain a result code of 0, the private port as indicated in the deletion request, a public port of 0, and a lifetime of 0. The NAT gateway MUST respond to a request to destroy a mapping that does not exist as if the request were successful. This is because of the case where the acknowledgement is lost, and the client retransmits its request to delete the mapping. In this case the second request to delete the mapping MUST return the same response packet as the first request.

If the deletion request was unsuccessful, the response MUST contain a non-zero result code and the requested mapping; the lifetime is undefined (MUST be set to zero on transmission, and MUST be ignored on reception). If the client attempts to delete a port mapping which was manually assigned by some kind of configuration tool, the NAT gateway MUST respond with a 'Not Authorized' error, result code 2.

When a mapping is destroyed as a result of its lifetime expiring or for any other reason, if the NAT gateway's internal state indicates that there are still active TCP connections traversing that now-defunct mapping, then the NAT gateway SHOULD send appropriately-constructed TCP RST (reset) packets both to the local client and to the remote peer on the Internet to terminate that TCP connection.

A client can request the explicit deletion of all its UDP or TCP

mappings by sending the same deletion request to the NAT gateway with public port, private port, and lifetime set to 0. A client MAY choose to do this when it first acquires a new IP address in order to protect itself from port mappings that were performed by a previous owner of the IP address. After receiving such a deletion request, the gateway MUST delete all its UDP or TCP port mappings (depending on the opcode). The gateway responds to such a deletion request with a response as described above, with the private port set to zero. If the gateway is unable to delete a port mapping, for example, because the mapping was manually configured by the administrator, the gateway MUST still delete as many port mappings as possible, but respond with a non-zero result code. The exact result code to return depends on the cause of the failure. If the gateway is able to successfully delete all port mappings as requested, it MUST respond with a result code of 0.

3.4 Result Codes

Currently defined result codes:
0 - Success
1 - Unsupported Version
2 - Not Authorized/Refused
(e.g. box supports mapping, but user has turned feature off)
3 - Network Failure
(e.g. NAT box itself has not obtained a DHCP lease)
4 - Out of resources
(NAT box cannot create any more mappings at this time)
5 - Unsupported opcode

If the result code is non-zero, the format of the packet following the result code may be truncated. For example, if the client sends a request to the server with an opcode of 17 and the server does not recognize that opcode, the server SHOULD respond with a message where the opcode is 17 + 128 and the result code is 5 (opcode not supported). Since the server does not understand the format of opcode 17, it may not know what to place after the result code. In some cases, relevant data may follow the opcode to identify the operation that failed. For example, a client may request a mapping but that mapping may fail due to resource exhaustion. The server SHOULD respond with the result code to indicate resource exhaustion (4) followed by the requested port mapping so the client may identify

which operation failed.

Clients MUST be able to properly handle result codes not defined in this document. Undefined results codes MUST be treated as fatal errors of the request.

3.5 Recreating Mappings On Reboot

The NAT gateway MAY store mappings in persistent storage so when it is powered off or rebooted, it remembers the port mapping state of the network. However, maintaining this state is not necessary. When the NAT gateway powers on or clears its port mapping state as the result of a configuration change, it MUST reset the epoch time and re-announce its IP address as described in Section 3.2.1 "Announcing Address Changes". This will signal to clients on the network that they need to redo their mappings. When a client renews its port mappings as the result of receiving such a packet, it MUST delay each port mapping request by a random amount of time selected with uniform random distribution in the range 0 to 3 seconds.

3.6 Seconds Since Start of Epoch

Every packet sent by the NAT gateway includes a "Seconds since start of epoch" field (SSSOE). If the NAT gateway resets or loses the state of its port mapping table, due to reboot, power failure, or any other reason, it MUST reset its epoch time and begin counting SSSOE from 0 again. Whenever a client receives any packet from the NAT gateway, either gratuitously or in response to a client request, the client computes its own conservative estimate of the expected SSSOE value by taking the SSSOE value in the last packet it received from the gateway and adding 7/8 (87.5%) of the time elapsed since that packet was received. If the SSSOE in the newly received packet is less than the client's conservative estimate, then the client concludes that the NAT gateway has undergone a reboot or other loss of port mapping state, and the client MUST immediately renew all its active port mapping leases as described in Section 3.6 "Recreating

Mappings On Reboot".

4. UNSAF Considerations

The document "IAB Considerations for UNSAF Across NAT" [RFC 3424] covers a number of issues when working with NATs. RFC 3424 outlines some requirements for any document that attempts to work around problems associated with NATs. This section addresses those requirements.

4.1 Scope

This protocol addresses the needs of TCP and UDP transport peers that are separated from the public internet by exactly one NAT. Such peers must have access to some form of directory server for registering the public IP address and port at which they can be reached.

4.2 Transition Plan

Any client making use of this protocol SHOULD implement IPv6 support. If a client supports IPv6 and is running on a device with a global IPv6 address, that IPv6 address SHOULD be preferred to the IPv4 public address using this NAT mapping protocol. In case other clients do not have IPv6 connectivity, both the IPv4 and IPv6 addresses SHOULD be registered with whatever form of directory server is used. Preference SHOULD be given to IPv6 addresses when available. By implementing support for IPv6 and using this protocol for IPv4, vendors can ship products today that will work under both scenarios. As IPv6 is more widely deployed, clients of this protocol following these recommendations will transparently make use of IPv6.

4.3 Failure Cases

Aside from NATs that do not implement this protocol, there are a number of situations where this protocol may not work.

4.3.1 NAT Behind NAT

Some people's primary IP address, assigned by their ISP, may itself be a NAT address. In addition, some people may have a public IP address, but may then double NAT themselves, perhaps by choice or perhaps by accident. Although it might be possible in principle for one NAT gateway to recursively request a mapping from the next one, this document does not advocate that and does not try to prescribe how it would be done.

It would be a lot of work to implement nested NAT port mapping correctly, and there are a number of reasons why the end result might not be as useful as we might hope. Consider the case of an ISP that offers each of its customers only a single NAT address. This ISP could instead have chosen to provide each customer with a single public IP address, or, if the ISP insists on running NAT, it could have chosen to allow each customer a reasonable number of addresses, enough for each customer device to have its own NAT address directly from the ISP. If instead this ISP chooses to allow each customer just one and only one NAT address, forcing said customer to run nested NAT in order to use more than one device, it seems unlikely that such an ISP would be so obliging as to provide a NAT service that supports NAT Port Mapping Protocol. Supposing that such an ISP did wish to offer its customers NAT service with NAT-PMP so as to give them the ability to receive inbound connections, this ISP could easily choose to allow each client to request a reasonable number of DHCP addresses from that gateway. Remember that Net 10 [RFC 1918] allows for over 16 million addresses, so NAT addresses are not in any way in short supply. A single NAT gateway with 16 million available addresses is likely to run out of packet forwarding capacity before it runs out of private addresses to hand out. In this way the ISP could offer single-level NAT with NAT-PMP, obviating the need to support nested NAT-PMP. In addition, an ISP that is motivated to provide their customers with unhindered access to the Internet by allowing incoming as well as outgoing connections has better ways to offer this service. Such an ISP could offer its customers real public IP addresses instead of NAT addresses, or could even choose to offer its customers full IPv6 connectivity, where no mapping or translation is required at all.

4.3.2 NATs with Multiple Public IP Addresses

If a NAT maps private addresses to multiple public addresses, then it SHOULD pick one of those addresses as the one it will support for inbound connections, and for the purposes of this protocol it SHOULD act as if that address were its only address.

4.3.3 NATs and Routed Private Networks

In some cases, a large network may be subnetted. Some sites may install a NAT gateway and subnet the private network. Such subnetting breaks this protocol because the router address is not necessarily the address of the device performing NAT.

Addressing this problem is not a high priority. Any site with the resources to set up such a configuration should have the resources to add manual mappings or attain a range of globally unique addresses.

Not all NATs will support this protocol. In the case where a client is run behind a NAT that does not support this protocol, the software relying on the functionality of this protocol may be unusable.

4.3.4 Communication Between Hosts Behind the Same NAT

NAT gateways supporting NAT-PMP should also implement "hairpin translation". Hairpin translation means supporting communication between two local clients being served by the same NAT gateway.

Suppose device A is listening on private address and port 10.0.0.2:80 for incoming connections. Using NAT-PMP, device A has obtained a mapping to public address and port x.x.x.x:80, and has recorded this public address and port in a public directory of some kind. For example, it could have created a DNS SRV record containing this information, and recorded it, using DNS Dynamic Update [RFC 3007], in a publicly accessible DNS server. Suppose then that device B, behind the same NAT gateway as device A, but unknowing or uncaring of this fact, retrieves device A's DNS SRV record and attempts to open a TCP connection to x.x.x.x:80. The outgoing packets addressed to this public Internet address will be sent to the NAT gateway for translation and forwarding. Having translated the source address and port number on the outgoing packet, the NAT gateway needs to be smart enough to recognize that the destination address is in fact itself, and then feed this packet back into its packet reception engine, to perform the destination port mapping lookup to translate and forward this packet to device A at address and port 10.0.0.2:80.

4.3.5 Non UDP/TCP Transport Traffic

Any communication over transport protocols other than TCP and UDP will not be served by this protocol. Examples are Generic Routing Encapsulation (GRE), Authentication Header (AH) and Encapsulating Security Payload (ESP).

4.4 Long Term Solution

As IPv6 is deployed, clients of this protocol supporting IPv6 will be able to bypass this protocol and the NAT when communicating with other IPv6 devices. In order to ensure this transition, any client implementing this protocol SHOULD also implement IPv6 and use this solution only when IPv6 is not available to both peers.

4.5 Existing Deployed NATs

Existing deployed NATs will not support this protocol. This protocol will only work with NATs that are upgraded to support it.

5. Security Considerations

As discussed in section 3.2, only clients on the private side of the NAT may create port mappings, and only on behalf of themselves. By using IP address spoofing, it's possible for one client to delete the port mappings of another client. It's also possible for one client to create port mappings on behalf of another client. The best way to deal with this vulnerability is to use IPSec [RFC 2401].

Since allowing incoming connections is often a policy decision, any NAT gateway implementing this protocol SHOULD have an administration mechanism to disable it.

Some people view the property that NATs block inbound connections as a security benefit which is undermined by this protocol. The authors of this document have a different point of view. In the days before NAT, all hosts had unique public IP addresses, and had unhindered

ability to communicate with any other host on the Internet. When NAT came along it broke this unhindered connectivity, relegating many hosts to second-class status, unable to receive inbound connections. This protocol goes some way to undo some of that damage. The purpose of a NAT gateway should be to allow several hosts to share a single address, not to simultaneously impede those host's ability to communicate freely. Security is most properly provided by end-to-end cryptographic security, and/or by explicit firewall functionality, as appropriate. Blocking of certain connections should occur only as a result of explicit and intentional firewall policy, not as an accidental side-effect of some other technology.

6. Copyright Notice

7. IANA Considerations

No IANA services are required by this document.

8. Acknowledgments

The concepts described in this document have been explored, developed and implemented with help from Bob Bradley, Josh Graessley, Rob Newberry, Roger Pantos, John Saxton, and James Woodyatt.

9. References

[RFC 1918] Y. Rekhter et.al., "Address Allocation for Private Internets", RFC 1918, February 1996.

[RFC 2119] RFC 2119 - Key words for use in RFCs to Indicate Requirement Levels

[RFC 2131] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC 2401] Atkinson, R. and S. Kent, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC 2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC 3007] Wellington, B., "Simple Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

[RFC 3022] RFC 3022 - Network Address Translator

[RFC 3424] RFC 3424 - IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation

[ETEAISD] J. Saltzer, D. Reed and D. Clark: "End-to-end arguments in system design", ACM Trans. Comp. Sys., 2(4):277-88, Nov. 1984

10. Authors' Addresses

Stuart Cheshire
Apple Computer, Inc.
1 Infinite Loop
Cupertino
California 95014
USA

Phone: +1 408 974 3207
EMail: rfc [at] stuartcheshire [dot] org

Marc Krochmal
Apple Computer, Inc.
1 Infinite Loop
Cupertino
California 95014
USA

Phone: +1 408 974 4368
EMail: marc [at] apple [dot] com

Kiren Sekar
Apple Computer, Inc.
1 Infinite Loop
Cupertino
California 95014
USA

Phone: +1 408 974 8051
EMail: kiren [at] apple [dot] com