

Network Working Group
Internet-Draft
Expires: August 29, 2001

M. Stenberg
S. Paavolainen
T. Ylonen
T. Kivinen
SSH Communications Security Corp
February 28, 2001

IPsec NAT-Traversal
draft-stenberg-ipsec-nat-traversal-02.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 29, 2001.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

This draft details the changes needed in order to make both initial IKE negotiation and subsequent authenticated/encrypted communications across IPsec AH/ESP SAs work despite the changes in the headers, and possible protocol transformations.

Internet-Draft

IPsec NAT-Traversal

February 2001

Table of Contents

1.	Introduction	3
1.1	Definitions	3
2.	Analysis	4
2.1	Assumptions	4
2.2	IPsec cases	4
2.2.1	Host-to-host	5
2.2.2	Host-to-network	5
2.2.3	Network-to-network	5
2.3	Issues stemming from NAT technology	5
2.4	Summary of issues	5
3.	Solution	7
3.1	IKE probe	7
3.1.1	Determining of support	7
3.1.2	NAT-Traversal need-probe	8
3.2	IPsec SA traffic encapsulation	8
3.3	Keepalive	10
3.4	Built-in NAT	10
4.	Known issues with the solution	12
4.1	Conceptual issues	12
4.2	Overhead	12
4.3	Security considerations	13
4.4	Intellectual property rights	13
	Authors' Addresses	14
	References	14
A.	Changes since previous version	16
B.	Implementation notes of de-/encapsulation	17
	Full Copyright Statement	19

Internet-Draft

IPsec NAT-Traversal

February 2001

1. Introduction

NAT devices have proliferated recently. Increased number of IPv6-enabled devices will not automatically mean disappearance of NAT devices, as IPv4 will be probably in use for decade(s). There will be need for bridging between IPv4 and IPv6 networks, and as long as there are NATs around, basic IPsec as defined by [RFC 2401](#) [1] will not work. It is quite important that there is a defined standard for handling IPsec traffic in networks with NAT devices. Preferably, a standard will evolve to fit all possible cases that may arise.

In [Section 2](#), most of the different IPsec+NAT permutations are analyzed and a list of issues is presented. [Section 3](#) details the proposed solution to these issues. Finally, potential problems in the solution are noted in [Section 4](#).

1.1 Definitions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC 2119](#) [2].

NAT terminology used is described in [RFC 2663](#) [3], with the following exception:

- o Protocol NAT: Protocol NAT is a NAT process/device that changes the protocol of the packet; this usually involves a whole new header for the packet.

[2. Analysis](#)

[2.1 Assumptions](#)

It can be safely assumed that IKE, as defined in [RFC 2409](#) [4], works. IKE negotiations are handled with normal UDP traffic. Therefore, it should work despite network address changes along the route. IP packet payloads are assumed to be left unmodified; changes to the UDP headers can occur, as long as nothing drops the packets before they reach the host.

As normal IPsec traffic does not pass across NATs (nor protocol NATs), a complete NAT-Traversal design should encapsulate IPsec SAs in UDP packets, which behave like IP packets, but as they are used in applications, they can pass through all types of NATs.

[2.2 IPsec cases](#)

Initially, most of the different IPsec+NAT combinations are listed here to make sure that all implications of NAT use are addressed. IPsec cases can be divided into three different categories (with possible NATs in various places along the route between hosts employing IPsec).

- o Host-to-host (tunnel or transport mode)
- o Host-to-network (tunnel mode)

- o Network-to-network (tunnel mode)

In all cases, the IKE responder must be only behind a series of basic NATs with static address assignment. Dynamic address assignment does not work for obvious reasons (the IKE initiator cannot contact such an address). NATs do not work because most IKE implementations seem to be hardcoded to use port 500.

The IKE initiator can be behind any kind of NAT, although in cases where initiation of traffic from both directions should be allowed (primarily VPN-like cases), the same restrictions that apply to the responder apply also to the initiator.

Issue #1: Both hosts need to know that there is a NAT in the middle, but currently IKE/IPsec do not provide such method. Only indication of NAT presence is the fact that all IPsec SA packets, if they arrive, will be dropped as invalid if AH is in use.

Issue #2: It is obvious that programs residing on an IKE responder that is behind a basic NAT cannot know about the existence of the NAT or about the specific address mappings configured there.

Therefore the IKE responder implementation should have advance knowledge about the address mappings.

[2.2.1](#) Host-to-host

Host-to-host traffic using tunnel or transport mode is the most basic case; it only becomes interesting if there is no shared address space between the parties (a VPN of sorts) and there are NATs in between.

Issue #3: If NATs are employed along the route, there may be addressing conflicts in tunnel mode (and there WILL be conflicts in transport mode). From the IKE responder point of view, the IKE initiators' addresses may conflict if they are in private networks (such as the IANA-assigned 10.0.0.0 subnet).

[2.2.2](#) Host-to-network

Only tunnel mode is applicable for host-to-network communication, and the only apparent problem is the potential lack of shared address space (a host without an address in the remote network that

it is accessing). Therefore, there is potential for issue #3-type of problems.

[2.2.3](#) Network-to-network

Only tunnel mode is applicable in network-to-network communication, and issue #3 is potentially also a problem. That is mostly outside scope of this draft, as at the moment such a case is rarely encountered.

[2.3](#) Issues stemming from NAT technology

Issue #4: The NATs with dynamic address assignment may change their address mapping suddenly (or they may be rebooted), making the remote host concept unworkable even as a unique ip-port pair.

[2.4](#) Summary of issues

There are basically four problems that need addressing:

1. detection of network address translation during IKE negotiation (issue #1),
2. a way of sending packets along the network so that NAT effects can be countered, yet the security of the system will not be affected (UDP encapsulation; assumption),
3. keeping NAT mapping static - NAT devices with dynamic address

assignment configurations typically contain timeouts that will cause changes in addressing, if not circumvented by using keepalive packets to maintain the specific mappings (issue #4), and

4. the lack of unique IP addresses in the NAT world; it is possible for a server to have several clients with the same configured IP address, although they appear to the server to be from different hosts/ports (issue #3).

Issue #2 (basic NAT case, where the IKE responder does not know what address to use) is easily solved, as seen in the end of [Section 3.2](#).

[3.](#) Solution

The solution that resolves all the issues mentioned in [Section 2.4](#) can be divided into four different parts, explained in this section:

- o IKE probe to detect NAT presence

- o IPsec SA traffic encapsulation to counter NAT effects
- o NAT translation keepalive messages, which maintain NAT mappings
- o built-in NAT (if needed) to make addresses unique

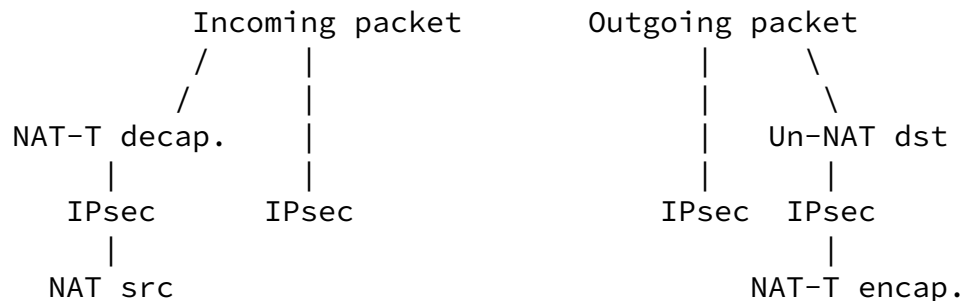


Figure 1: IPsec processing with and without a NAT-Traversal process.

[3.1](#) IKE probe

There is a need for two different exchanges during the IKE negotiation. Initially, it needs to be determined whether or not both sides support NAT-Traversal. Then, if both sides do support it, there should be a probe sequence that results in knowledge about whether or not the network between hosts contains a NAT device.

Although this involves four messages, it is possible to make this work in all modes, as seen shortly.

[3.1.1](#) Determining of support

The NAT-Traversal capability of the remote host is determined by an exchange of vendor strings; in Phase 1 two first messages, the vendor id payload for this specification of NAT-Traversal (MD5 hash of "[draft-stenberg-ipsec-nat-traversal-02](#)" - ['0x61', '0x5', '0xc4', '0x22', '0xe7', '0x68', '0x47', '0xe4', '0x3f', '0x96', '0x84', '0x80', '0x12', '0x92', '0xae', '0xcd']) MUST be sent if supported (and it MUST be received by remote side) for the NAT-Traversal probe to continue.

[3.1.2](#) NAT-Traversal need-probe

Once the NAT-Traversal support of both parties has been established in the initial stages of Phase 1, further inquiries need to be made using private payloads.

Initially, in the 5th message of Main Mode / 3rd message of Aggressive Mode, the initiator will add one private payload to the message. PAYLOAD_TYPE (from private range) is 211.

The payload should contain the following:

{perceived remote identity -
IP address and UDP port}

{one or more local identities -
local interface IP address+IKE UDP port numbers}

Figure 2: Probe payload in Main Mode message #5

The probe payload is encoded as a series of Identification Payloads of [RFC 2407](#) [6], with the perceived remote identity as the first payload, and the local identities as the following payloads.

Once the IKE responder receives the payload represented in Figure 2, the remote should check whether or not the remote identity, as perceived by the IKE initiator, matches one of the locally configured interface addresses (with proper port number). Also, the remote identity as perceived by the IKE responder should match one of the ip-port pairs sent in the packet.

If one (or two) of those tests fails, the responder knows that NAT-Traversal is needed. The decision on whether to use NAT-Traversal or not is left up to the responder, and the responder transmits the decision as a private payload of type 211 in the last message of Main Mode, or in the first or second message of Quick Mode (depending on who initiates, the first message from the responder may be either first or second).

The payload is one or more bytes long. Implementations conforming to this draft version should just examine the first byte. The byte should be 0x00 when NAT-Traversal should not be used. All other values indicate that NAT-Traversal should be used.

[3.2](#) IPsec SA traffic encapsulation

Automatic use of NAT-Traversal encapsulation for IKE-negotiated IPsec SAs MUST NOT be done. Instead, NAT-Traversal SHOULD be used only when IKE negotiation has resulted in a decision to use

Internet-Draft

IPsec NAT-Traversal

February 2001

NAT-Traversal, or when manually keyed IPsec SAs are configured to use it.

Traffic that is not in AH or ESP format MUST NOT be encapsulated using this scheme, as that would provide a way to create distributed denial of service attacks, and possibly also some other security threats.

Normal AH/ESP traffic does not pass through NATs unmodified. Typically, the source or destination address may change, which makes the resulting AH/ESP packet unusable. Thus, there has to be enough redundant data to be able to recreate a packet to its original form. To make the implementation simpler, it should follow the same NAT route as IKE packets.

Therefore (as noted before), the traffic has to be encapsulated as UDP packets between two hosts (which implies that they follow same route even in NATs) using the IKE port. The basic idea behind this NAT-Traversal data encapsulation format is that it should be a format that can be adapted to future needs. The only requirement for this initial version is that it contains a version number, and it is invalid for IKE purposes.

An IPsec NAT-Traversal envelope for IPv4 packet encapsulation looks like this:

<IP HEADER:

src=local IP address

dst=perceived remote host>

<UDP header:

srcport=500 (local IKE port),

dstport=perceived remote port>

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
+-----+-----+-----+-----+																															
								IKE initiator cookie - 4 first bytes (00000000)																							
+-----+-----+-----+-----+																															
								IKE initiator cookie - 4 last bytes (00000000)																							
+-----+-----+-----+-----+																															
NAT-T IP4 hlen reserved - 0																IP4 ID															
+-----+-----+-----+-----+																															

<IP4 HEADER options, if any>

Figure 3: A NAT-Traversal envelope for an IPv4 IPsec packet.

IKE initiator cookie that contains only zeros is used only

between consenting NAT-Traversal implementations that conform to this draft (normal IKE conversations should never involve IKE initiator cookie that consists of only zeros, as shown in [section 2.4 of RFC 2408](#) [5]).

NAT-T field is lower 4 bits of the first payload byte. It contains the IP address type and the IP protocol used, as follows: IPv4-AH: 1, IPV4-ESP: 2.

IP4 hlen is the original header length field, which includes the IP4 header options.

IP4 ID is the original IP4 packet Identification.

Description of how the IPsec encapsulation and decapsulation should be implemented is in the [Appendix B](#).

[3.3](#) Keepalive

Disclaimer: the IKE SA heartbeat should probably be used whenever one becomes a standard. Until then, NAT-Traversal will have its own keepalive packets that are entirely separate from the IKE SA.

The sole purpose of the keepalives is to keep the NATs along the route between hosts from removing the mapping from their dynamic configuration (if any). Therefore, the actual contents of the keepalive packets can be more or less ignored (unless they stop arriving), and thus encrypting them would serve no useful purpose.

Keepalive packets MUST be sent as long as there is at least one IKE-probed IPsec SA in existence between two hosts that employ NAT-Traversal to communicate with each other. Both sides MUST send a keepalive packet every KEEPALIVE_INTERVAL (=9) seconds. The 9 was picked as reasonable compromise with assumption that nobody would be insane enough to set less than 30 second timeout for NAT mappings (30 seconds exists out there). 9 second keepalive requires 3

sequential keepalive messages to be lost in order for the NAT to lose it's mapping.

If no keepalive packets from remote side are received for a while, implementation SHOULD consider the connection dead and drop the IPsec SAs prematurely. Specific period can vary by implementation. Typically some multiple of KEEPALIVE_INTERVAL + some value is reasonable, f.ex. $5 * \text{KEEPALIVE_INTERVAL} + 3$ (in order to make the SAs time out if 5 sequential keepalives are lost).

[3.4](#) Built-in NAT

Built-in basic NAT implementation within the IPsec stack is

Stenberg, et. al.

Expires August 29, 2001

[Page 10]

Internet-Draft

IPsec NAT-Traversal

February 2001

necessary in some tunnel cases and all transport cases. To stay consistent with [RFC 2409](#) [4], which specifies that both tunnel and transport mode MUST be supported, we define that there MUST be a built-in basic NAT implementation for NAT-Traversal use.

The built-in NAT is needed in some cases where issue #3 surfaces (see [Section 2.2](#) for details) to make the remote host(s) unique. Typically, the host mapping should be from perceived remote_host-remote_port to some internal A- or B-class network. Whenever the remote side successfully initiates IPsec SA employing NAT-Traversal, there should be an internal NAT definition for the (remote_host, remote_port) if one is required according to the local configuration (or if transport mode is used, in which case internal basic NAT SHOULD always be employed). Whenever IPsec processing for an incoming packet is done, the internal basic NAT should be done to the src. Whenever an outgoing packet headed towards an internal NAT address enters the IPsec, the internal NAT address should be changed to the address that was used for negotiating the IPsec SA.

In tunnel mode, it is possible that entire networks may need masking. In the NAT-Traversal+IPsec case, a separate NAT box would not know about the perceived remote_host-remote_port pair which provides uniqueness to the tunneled IP addresses. Therefore, there is a need for NAT within the IPsec implementation. This MAY be supported, but specific implementation details are not provided in this draft.

[4.](#) Known issues with the solution

[4.1](#) Conceptual issues

The non-unique hosts may cause problems, as there is a potential problem of ip-port-proto-spi not being unique any more. The problem does not surface in the incoming traffic, but it may occur in the outgoing traffic case. There are (at least) a couple of different solutions to the problem:

- o Tying the remote-host,remote-port of NAT-T IPsec SA decapsulation and the ip-port-proto-spi.
- o Refusal of duplicate IPsec SA SPIs during IKE P2QM negotiation.
- o SAD may be extended to use (remote-perceived-ike-peer, ip, proto, spi) as unique key instead of (ip, proto, spi).

[4.2](#) Overhead

This solution is about the most minimal possible that covers the eventual possibilities (reasonable combinations of AH, ESP and

IPCOMP) without becoming overly complex. Different types of overhead caused by this draft are noted here, as well as possible ways of decreasing/removing the overheads involved. Processing time and memory overhead are ignored as negligible (some more processing for each packet, potentially logarithmic searches for free addresses, minimal extra data for each IPsec SA).

- o IKE P1 negotiation extra payloads: Moderately small, typically less than 200 bytes. It appears that this cannot be reduced.
- o Each IPv4-based IPsec SA packet will contain extra overhead of 20 bytes (8 bytes of UDP header, 12 bytes of NAT-T header). The impact of the additions is not typically great; for example, ethernet's minimum packet length will cover this for minimal length packets, and for slow networks there may be protocols such as PPP that provides header- or even whole payload compression (in that case, the exactly same UDP, NAT-T- and start of ESP/AH-header should cause negligible overhead).
- o Keepalive overhead of 56 bytes every KEEPALIVE_INTERVAL (20 bytes of IP header + 8 bytes of UDP header = 28 bytes times 2 for bidirectional traffic). 6 bytes/second may seem excessive, but as long as a general-purpose solution is desired, it cannot be bypassed. Two consenting parties that know there is only static NATs in-between MAY, of course, skip heartbeats altogether.

[4.3](#) Security considerations

Whenever changes to some fundamental parts of a security protocol are proposed, the examination of security implications cannot be skipped. Therefore, here are some observations on the effects, and whether or not these effects matter. This section will be expanded further in future versions of this draft.

- o IKE probe reveals NAT-Traversal support to everyone. This should not be an issue.
- o The value of authentication mechanisms based on IP addresses disappears once NATs are in the picture. That is not necessarily a bad thing (for any real security, other authentication measures than IP addresses should be used).

- o Some DoS implications exist; a single malicious user can possibly allocate up to (number-of-hosts-available) * 65535 (=number-of-ports-on-host) internal host IP addresses at the same time - and cause that many negotiations (this is 65535 times as much DoS potential as traditional IKE). As the IP addresses are allocated only after authentication is successful, the attacker is known. Therefore this can be considered only a slight risk, as it can be ameliorated by adding allocations-per-remote-end-entity limits.
- o Although two last packets in the Main Mode are encrypted, the IKE responder (if improper) gets some internal IP address information that IKE initiator might not want to reveal.

[4.4](#) Intellectual property rights

SSH Communications Security Corp hereby announces that one or more patents or patent applications may be relevant to this internet-draft. If this internet-draft becomes an IETF standard, SSH Communications Security Corp intends to support a widespread adoption of the standard by offering - on the basis of reciprocity whenever applicable - to license any IPR owned by SSH Communications Security Corp necessary for implementing the standard on fair, reasonable and nondiscriminatory terms.

References

- [1] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.

- [3] Srisuresh, S. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", [RFC 2663](#), August 1999.
- [4] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", [RFC 2409](#), November 1998.
- [5] Maughan, D., Schertler, M., Schneider, M. and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", [RFC 2408](#), November 1998.
- [6] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", [RFC 2407](#), November 1998.

Authors' Addresses

Markus Stenberg
SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 Helsinki
Finland

EMail: mstenber@ssh.com

Santeri Paavolainen
SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 Helsinki
Finland

EMail: santtu@ssh.com

Tatu Ylonen
SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 Helsinki
Finland

EMail: ylo@ssh.com

SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 Helsinki
Finland

E-Mail: kivinen@ssh.com

[Appendix A](#). Changes since previous version

In addition some small fixes to the document (minor changes in wording), there are some significant changes in this revision as compared to the previous one.

Aggressive mode support was added by making it possible to have final NAT-T decision private payload sent in QM packet.

IPcomp as outer header was removed (due to conflict with goals specified in [Section 3.2](#)).

ToS field was removed as it isn't required by AH and not really useful to encapsulate regardless.

[Appendix B](#). Implementation notes of de-/encapsulation

Note that the IP checksum needs to be updated constantly, or alternatively verified in the beginning and recalculated in the end of both decapsulation and encapsulation.

Encapsulation should happen after IPsec processing and work as follows (with data changing as shown in Figure 4):

1. Insert UDP and NAT-T headers to the end of minimal IP4 header (offset of 20 bytes from beginning of the packet). UDP data: srcport=local-ike-port, dstport=perceived-remote-port.
2. Remove the excess IP4 header bytes from the checksum (NAT-T uses only minimal IP4 header - 20 bytes).
3. Store IP4 original header length in the NAT-T header. Set NAT-T field according to protocol of the IP packet. Identification should be copied as-is.
4. Set the IP4 header length to 20 bytes.
5. Set the IP4 protocol to be UDP.

IP [N bytes]
AH/ESP
...

to

minimal-IP [20 bytes]
UDP [8 bytes]
NAT-T [4 bytes]
IP-header-options [N-20 bytes]
AH/ESP

Figure 4: NAT-Traversal encapsulation process.

Decapsulation should happen before IPsec processing and work as follows (and work like reverse of Figure 4):

1. Check that protocol is UDP and dstport == IKE port.
2. If the packet is keepalive (empty UDP payload), update reachability data, if any, and drop the packet.
3. Check that initiator cookie is zeros - pass if not (normal IKE

content).

4. Note the remote ip-port pair and look up respective IKE/IPsec data - drop if unsuccessful.
5. Copy header length and identification from NAT-T header to the IP packet.
6. Set IP protocol according to NAT-T type.
7. Change the source and destination address to be the IPsec endpoints involved (using either SPI or alternatively tying the perceived remote_ip-remote_host to single src-dst pair).
8. Eliminate the UDP and NAT-T headers from middle of the packet.

Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC editor function is currently provided by the
Internet Society.