IP Security Protocol Working Group (IPSEC)
INTERNET-DRAFT
draft-ietf-ipsec-nat-t-ike-01.txt
Expires: 23 April 2001

T. Kivinen, M. Stenberg
SSH Communications Security
A. Huttunen
F-Secure Corporation
W. Dixon, B. Swander
Microsoft
V. Volpe
Cisco Systems
L. DiBurro
Nortel Networks
23 October 2001

Negotiation of NAT-Traversal in the IKE

Status of This Memo

Abstract

This document describes how to detect one or more NATs between IPsec hosts, and how to negotiate the use of UDP encapsulation of the IPsec packets through the NAT boxes in IKE.

Table of Contents

1. Introduction

This document is split in two parts. The first part describes what is needed in the IKE phase 1 for the NAT-Traversal support. This includes detecting if the other end supports NAT-Traversal, and detecting if there is one or more NAT along the path from host to host.

The second part describes how to negotiate the use of UDP encapsulated IPsec packets in the IKE Quick Mode. It also describes how to transmit the original source address to the other end if needed. The original source address can be used to incrementally update the TCP/IP checksums so that they will match after the NAT transform (The NAT cannot do this, because the TCP/IP checksum is inside the UDP encapsulated IPsec packet).

The document [Hutt01] describes the details of the UDP encapsulation and the document [Dixon01] provides background information and motivation of the NAT-Traversal in general.

2. Specification of Requirements

This document shall use the keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED, "MAY", and "OPTIONAL" to describe requirements. They are to be interpreted as described in [RFC-2119] document.

3. Phase 1

The detection of the support for the NAT-Traversal and detection of the NAT along the path happens in the IKE [RFC-2409] phase 1.

The NAT is supposed to float the IKE UDP port, and recipients MUST be able to process IKE packets whose source port is different than 500. There are cases where the NAT does not have to float the source port (only one (IPsec) host behind the NAT or for the first IPsec host it can

keep the port 500, and float only the following IPsec hosts).

Recipients MUST reply back to the source address from the packet. This also means that when the original responder is doing rekeying, or sending notifications etc. to the original initiator it MUST send the packets from the same set of port and IP numbers that was used when the IKE SA was originally created (i.e the source and destination port and IP numbers must be same).

For example the initiator sends packet having source and destination port 500, the NAT changes that to such packet which have source port 12312 and destination port 500, the responder must be able to process the packet whose source address is that 12312 and it must reply back with packet whose source address is 500 and destination address 12312, the NAT will then translate this packet to have source address 500 and destination address 500.

3.1. Detecting support of Nat-Traversal

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-ietf-ipsec-nat-t-ike-00" - ["4485152d 18b6bbcd 0be8a846 9579ddcc"]) MUST be sent if supported (and it MUST be received by both sides) for the NAT-Traversal probe to continue.

3.2. Detecting presense of NAT

The purpose of the NAT-D payload is twofold, It not only detects the presence of NAT between two IKE peers, it also detects where the NAT is. The location of the NAT device is important in that the keepalives need to initiate from the peer "behind" the NAT.

To detect the NAT between the two hosts, we need to detect if the IP address or the port changes along the path. This is done by sending the hashes of IP address and port of both source and destination addresses from each end to another. When both ends calculate those hashes and get same result they know there is no NAT between. If the hashes do not match, somebody translated the address or port between, meaning we need to do NAT-Traversal to get IPsec packet through.

If the sender of the packet does not know his own IP address (in case of multiple interfaces, and implementation don't know which is used to route the packet out), he can include multiple local hashes to the packet (as separate NAT-D payloads). In this case the NAT is detected if and only if none of the hashes match.

The hashes are sent as a series of NAT-D (NAT discovery) payloads. Each payload contains one hash, so in case of multiple hashes, multiple NAT-D payloads are sent. In normal case there is only two NAT-D payloads.

The NAT-D payloads are included in the third and fourth packet in the main mode and second and third packet in the aggressive mode.

The format of the NAT-D packet is

```
 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
+---------------+---------------+---------------+---------------+
| Next Payload  |    RESERVED   |        Payload length         |
+---------------+---------------+---------------+---------------+
~               HASH of the address and port                    ~
+---------------+---------------+---------------+---------------+
```

The payload type for the NAT discovery payload is 130 (XXX CHANGE).

The HASH is calculated as follows:

        HASH = HASH(CKY-I | CKY-R | IP | Port)

using the negotiated HASH algorithm. All data inside the HASH is in the network byte-order. The IP is 4 octets for the IPv4 address and 16 octets for the IPv6 address. The port number is encoded as 2 octet

number in network byte-order. The first NAT-D payload contains the remote ends IP address and port (i.e the destination address of the UDP packet). The rest of the NAT-D payloads contain possible local end IP addresses and ports (i.e all possible source addresses of the UDP packet).

If there is no NAT between then the first NAT-D payload should match one of the local NAT-D packet (i.e the local NAT-D payloads this host is sending out), and the one of the other NAT-D payloads must match the remote ends IP address and port. If the first check fails (i.e first NAT-D payload does not match any of the local IP addresses and ports), then it means that there is dynamic NAT between, and this end should start sending keepalives as defined in the [Hutt01].

The CKY-I and CKY-R are the initiator and responder cookies, and they are added to the hash to make precomputation attacks for the IP address and port impossible.

An example of phase 1 exchange using NAT-Traversal in main mode (authentication with signatures) is:

```
         Initiator                           Responder
        ------------                        ------------
        HDR, SA, VID                   -->
                                       <-- HDR, SA, VID
        HDR, KE, Ni, NAT-D, NAT-D      -->
                                       <-- HDR, KE, Nr, NAT-D, NAT-D
        HDR*, IDii, [CERT, ] SIG_I     -->
                                       <-- HDR*, IDir, [ CERT, ], SIG_R
```

An example of phase 1 exchange using NAT-Traversal in aggressive mode (authentication with signatures) is:

```
         Initiator                           Responder
        ------------                        ------------
```



```
        HDR, SA, KE, Ni, IDii, VID     -->
                                       <-- HDR, SA, KE, Nr, IDir, [CERT, ],
                                             VID, NAT-D, NAT-D, SIG_R
        HDR*, [CERT, ], NAT-D, NAT-D,
          SIG_I                        -->
```

4. Quick Mode

After the Phase 1 both ends know if there is a NAT present between. The final decision of using the NAT-Traversal is left to the quick mode. The use of NAT-Traversal is negotiated inside the SA payloads of the quick mode. In the quick mode both ends can also send the original source addresses of the IPsec packets (in case of the transport mode) to the other, end so the other end has possibility to fix the TCP/IP checksum field after the NAT transform.

This sending of the original source address is optional, and it is not useful in the UDP-Encapsulated-Tunnel mode, as there is going to be proper IP header inside the UDP-Encapsulated packet. In case of only UDP-Encapsulated-Tunnel mode is negotiation then both ends SHOULD NOT send original source address.

It also might be unnecessary in the transport mode if the other end can turn off TCP/IP checksum verification. If the sending end knows (for example from the vendor id payload) that the other end can turn off TCP/IP checksum verification, he MAY leave the original source address payload away. Otherwise he SHOULD send the original source address.

4.1. Negotiation of the NAT-Traversal encapsulation

The negoation of the NAT-Traversal happens by adding two new encapsulation modes. These encapsulation modes are:

```
UDP-Encapsulated-Tunnel         61443 (XXX CHANGE)
UDP-Encapsulated-Transport      61444 (XXX CHANGE)
```

It is not normally useful to propose both normal tunnel or transport mode and UDP-Encapsulated modes. If there is a NAT box between normal tunnel or transport encapsulations may not work, and if there is no NAT box between, there is no point of wasting bandwidth by adding UDP encapsulation of packets. Because of this initiator SHOULD NOT include both normal tunnel or transport mode and UDP-Encapsulated-Tunnel or UDP-Encapsulated-Transport in its proposals.

4.2. Sending the original source address

In case of transport mode both ends SHOULD send the original source address to the other end. For the tunnel mode both ends SHOULD NOT send original source address to the other end.

The original source address of packets put to this transport mode IPsec SA is sent to other end using NAT-OA (NAT Original Address) payload.

The NAT-OA payloads are sent inside the first and second packets of the quick mode. The initiator SHOULD send the payload if it proposes any UDP-Encapsulated-Transport mode and the responder SHOULD send the payload only if it selected UDP-Encapsulated-Transport mode. I.e it is possible that initiator send the NAT-OA payload, but proposes both UDP-Encapsulated transport and tunnel mode, and then the responder selectes the UDP-Encapsulated tunnel mode and do not send NAT-OA payload back.

A peer MUST NOT fail a negotiation if it does not receive a NAT-OA payload if the NAT-OA payload only would contain redundant information. I.e. only the machine(s) that are actually behind the NAT need to send the NAT-OA payload. A machine with a public, non-changing IP address doesn't need to send the NAT-OA payload.

The format of the NAT-OA packet is

```
  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
 +---------------+---------------+---------------+---------------+
 | Next Payload  |   RESERVED    |        Payload length         |
 +---------------+---------------+---------------+---------------+
 |    ID Type    |   RESERVED    |           RESERVED            |
 +---------------+---------------+---------------+---------------+
 |         IPv4 (4 octets) or IPv6 address (16 octets)           |
 +---------------+---------------+---------------+---------------+
```

The payload type for the NAT discovery payload is 131 (XXX CHANGE).

The ID type is defined in the [RFC-2407]. Only ID_IPV4_ADDR and ID_IPV6_ADDR types are allowed. The two reserved fields after the ID Type must be zero.

An example of quick mode using NAT-OA payloads is:

```
 Initiator                        Responder
------------                     ------------
HDR*, HASH(1), SA, Ni, [, KE]
  [, IDci, IDcr ] [, NAT-OA] -->
                              <-- HDR*, HASH(2), SA, Nr, [, KE]
                                    [, IDci, IDcr ] [, NAT-OA]
HDR*, HASH(3)
```

5. 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). This means that pre-shared-keys authentication cannot be used with the main mode without group shared keys for everybody behind the NAT box, which is huge security risk.

o As the internal address space is only 32 bits, and it is usually very sparce, it might be possible for the attacker to find out the internal address used behind the NAT box by trying all possible IP-addresses and trying to find the matching hash. The port numbers are normally fixed to 500, and the cookies can be extracted from the packet. This limits the hash calculations down to 2^32. If educated guess of use of private address space is done, then the number of hash calculations needed to find out the internal IP address goes down to the 2^24 + 2 * (2^16).

o The NAT-D payloads nor the Vendor ID payloads are not authenticated at all in the main mode nor in the aggressive mode. This means that attacker can remove those payloads, modify them or add them. By removing or adding them the attacker can cause Denial Of Service attacks. By modifying the NAT-D packets the attacker can cause both ends to use UDP-Encapsulated modes instead of directly using tunnel or transport mode, thus wasting some bandwidth.

o The sending of the original source address in the Quick Mode releveas the internal ip address behind the NAT to the other end. In this case we have already authenticated the other end, and sending of the original source address is only needed in transport mode.

6. Intellectual property rights

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

SSH Communications Security Corp has notified the working group of one or more patents or patent applications that may be relevant to this internet-draft. SSH Communications Security Corp has already given a

licence for those patents to the IETF. For more information consult the online list of claimed rights.

7. Acknowledgments

Thanks to Tatu Ylonen, Santeri Paavolainen, and Joern Sierwald who contributed to the drafts used as base for this document.

8. References

[RFC-2409] Harkins D., Carrel D., "The Internet Key Exchange (IKE)", November 1998

[RFC-2407] Piper D., "The Internet IP Security Domain Of Interpretation for ISAKMP", November 1998

[RFC-2119] Bradner, S., "Key words for use in RFCs to indicate Requirement Levels", March 1997

[Hutt01] Huttunen, A. et. al., "UDP Encapsulation of IPsec Packets", draft-ietf-ipsec-udp-encaps-01.txt, October 2001

[Dixon01] Dixon, W. et. al., "IPSec over NAT Justification for UDP Encapsulation", draft-ietf-ipsec-udp-encaps-justification-00.txt, June 2001

9. Authors' Addresses

Tero Kivinen
SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 HELSINKI
Finland
E-mail: kivinen@ssh.fi

Markus Stenberg
SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 HELSINKI
Finland
E-mail: mstenber@ssh.com

Ari Huttunen
F-Secure Corporation

Tammasaarenkatu 7,
FIN-00181 HELSINKI
Finland
E-mail: Ari.Huttunen@F-Secure.com

William Dixon
Microsoft
One Microsoft Way
Redmond WA 98052
E-mail: wdixon@microsoft.com

Brian Swander
Microsoft
One Microsoft Way
Redmond WA 98052
E-mail: briansw@microsoft.com

Victor Volpe
Cisco Systems
124 Grove Street
Suite 205
Franklin, MA 02038
E-mail: vvolpe@cisco.com

Larry DiBurro

Nortel Networks
80 Central Street
Boxborough, MA 01719
ldiburro@nortelnetworks.com