IP Security Protocol Working Group (IPSEC) T. Kivinen
INTERNET-DRAFT SSH Communications Security
draft-ietf-ipsec-nat-t-ike-03.txt A. Huttunen
Expires: 24 December 2002 F- Secure Corporation
B. Swander
Microsoft
V. Volpe
Cisco Systems
24 June 2002

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 [Hutt02] 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 last time used (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-03" - ["7d9419a6 5310ca6f 2c179d92 15529d56"]) MUST be sent if supported (and it MUST be received by both sides) for the NAT-Traversal probe to continue.

3.2. Detecting presence 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 [Hutt02].

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                      -->
```

The '#' sign identifies that those packets are sent to the floated port if NAT is detected.

4. Floating to the new ports

IPsec-aware NATs can cause problems. Some NATs will not float IKE source port 500 even if there are multiple clients behind the NAT. They can also map IKE cookies to demultiplex traffic instead of using the source port. Both of these are problematic for generic NAT transparency since it is difficult for IKE to discover the capabilities of the NAT. The best approach is to simply move the IKE traffic off port 500 as soon as possible to avoid any IPsec-aware NAT special casing. Moving IKE from port 500 to port 4500 is known as port floating.

Take the common case of the initiator behind the NAT. The initiator must quickly float to 4500 once the NAT has been detected to minimize the window of IPsec-aware NAT problems.

In main mode, the initiator MUST float on the ID payload if there is NAT between the hosts. The initiator MUST set both UDP source and destination ports to 4500. All subsequent packets sent to this peer (including informationals) MUST be sent on 4500. In addition, the IKE data MUST be prepended with a non-ESP marker allowing for demultiplexing of traffic as defined in [Hutt02].

Thus, the IKE packet now looks like:

```
        IP UDP(4500,4500) <non-ESP marker> HDR*, IDii, [CERT, ] SIG_I
```

assuming authentication using signatures. The 4 bytes of non-ESP marker is defined in the [Hutt02].

When the responder gets this packet he performs the usual decryption and processing of the various payloads. If this is successful, he MUST update local state so that all subsequent packets (including informationals) to the peer use the new port, and possibly new IP address obtained from the incoming valid packet. The port will generally be different since the NAT will map UDP(500,500) to UDP(X,500), and UDP(4500,4500) to UDP(Y,4500). The IP address will seldom be different from the pre-float IP address. The responder MUST respond with all subsequent IKE packets to this peer using UDP(4500,Y).

Similarly, if the responder needs to rekey the phase 1 SA, then he MUST start the negotiation using UDP(4500,Y). Any implementation that supports NAT traversal, MUST support negotiations that begin on port 4500. **If a negotiation starts on 4500, then it doesn't need to float** anywhere else in the exchange.

Once floating has occurred, if a packet is received on 500, that packet is old. If the packet is an informational, it MAY be processed if local policy allows. If the packet is a main mode or aggressive mode packet, it SHOULD be discarded.

Here is an example of phase 1 exchange using NAT-Traversal in main mode (authentication with signatures) with port floating:

```
         Initiator                        Responder
        ------------                     ------------
        UDP(500,500) HDR, SA, VID    -->
                                     <-- UDP(500,X) HDR, SA, VID
        UDP(500,500) HDR, KE, Ni,
                     NAT-D, NAT-D    -->
                                     <-- UDP(500,X) HDR, KE, Nr,
                                                    NAT-D, NAT-D
        UDP(4500,4500) HDR*#, IDii,
                      [CERT, ]SIG_I  -->
                                     <-- UDP(4500,Y) HDR*#, IDir,
                                                   [ CERT, ], SIG_R
```

The floating algorithm for aggressive mode is very similar. After the NAT has been detected, the initiator sends: IP UDP(4500,4500) <4 bytes of non-ESP marker> HDR*, [CERT,], NAT-D, NAT-D, SIG_I The responder does similar processing to the above, and if successful, MUST update his internal IKE ports. The responder MUST respond with all subsequent IKE packets to this peer using UDP(4500,Y).

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

        UDP(500,500) HDR, SA, KE,
                     Ni, IDii, VID   -->
                                     <-- UDP(500,X) HDR, SA, KE,
                                                    Nr, IDir, [CERT, ],
                                                    VID, NAT-D, NAT-D,
                                                    SIG_R
        UDP(4500,4500) HDR*#, [CERT, ],
                       NAT-D, NAT-D,
                       SIG_I         -->

                                             <-- UDP(4500, Y) HDR*#, ...
```

While floating, the port in the ID payload in Main Mode/Aggressive Mode MUST be 0.

The most common case for the responder behind the NAT is if the NAT is simply doing 1-1 address translation. In this case, the initiator still floats both ports to 4500. The responder uses the identical algorithm as above, although in this case, Y will equal 4500, since no port translation is happening.

A different floating case involves out-of-band discovery of the ports to use. For instance, if the responder is behind a port translating NAT, and the initiator needs to contact it first, then the initiator will need to determine which ports to use, usually by contacting some other server. Once the initiator knows which ports to use to traverse the NAT, generally something like UDP(Z,4500), he initiates using these ports. This is similar to the responder rekey case above in that the ports to use are already known upfront, and no additional floating need take place.

Also the first keepalive timer starts after floating to new port, no keepalives are sent to the port 500 mapping.

5. 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.

5.1. Negotiation of the NAT-Traversal encapsulation

The negotiation 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.

5.2. 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 selects 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 original address 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)
```

6. Initial contact notifications

The source IP and port address of the INITIAL-CONTACT notification for the host behind NAT are not meaningful, so the IP and port numbers MUST NOT be used for the determine which IKE/IPsec SAs to remove. The ID payload sent from the other SHOULD be used instead. I.e when INITIAL-CONTACT notification is received from the other end, the receiving end SHOULD remove all the SAs associated with the same ID payload.

7. Recovering from the expiring NAT mappings

There are cases where NAT box decides to remove mappings that are still alive (for example, the keepalive interval is too long, or the NAT box is rebooted). To recover from those ends which are NOT behind NAT SHOULD use the last valid authenticated packet from the other end to determine which IP and port addresses the should be used. The host behind dynamic NAT MUST NOT do this as otherwise it opens DoS attack possibility, and there is no need for that, because the IP address or port of other host will not change (it is not behind NAT).
Keepalives cannot be used for this purposes as they are not authenticated, but any IKE authenticated IKE packet or ESP packet can be used to notice that the IP address or the port has changed.

8. 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.

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. Use of group shared keys is NOT RECOMMENDED.

o As the internal address space is only 32 bits, and it is usually very sparse, 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 Neither NAT-D payloads or Vendor ID payloads are 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.

- The sending of the original source address in the Quick Mode reveals 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.

- Updating the IKE SA / ESP UDP encapsulation IP addresses and ports for each valid authenticated packet can cause DoS in case we have attacker who can listen all traffic in the network, and can change the order of the packet and inject new packets before the packet he has already seen. I.e attacker can take the authenticated packet from the host behind NAT, change the packet UDP source or destination ports or IP addresses and sent it out to the other end before the real packet reaches there. The host not behind the NAT will update its IP address and port mapping and sends further traffic to wrong host or port. This situation is fixed immediately when the attacker stops modifying the packets as the first real packet will fix the situation back to normal. Implementations MAY print out warning every time the mapping is changed, as in normal case it should not happen that often.

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

10. Acknowledgments

Thanks to Markus Stenberg, Larry DiBurro and William Dixon who contributed actively to this document.

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

11. 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

[Hutt02] Huttunen, A. et. al., "UDP Encapsulation of IPsec Packets", draft-ietf-ipsec-udp-encaps-03.txt, June 2002

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

12. Authors' Addresses

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

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

--
kivinen@ssh.fi
SSH Communications Security http://www.ssh.fi/
SSH IPSEC Toolkit http://www.ssh.fi/ipsec/