IP Security Protocol Working Group (IPsec)
INTERNET-DRAFT SafeNet
draft-ietf-ipsec-nat-t-ike-08.txt
Expires: 10 July 2004 Microsoft

T. Kivinen
A. Huttunen
B. Swander
F-Secure Corporation
V. Volpe
Cisco Systems
10 Feb 2004

Negotiation of NAT-Traversal in the IKE

Status of This Memo

This document is a submission to the IETF IP Security Protocol (IPSEC) Working Group. Comments are solicited and should be addressed to the working group mailing list (ipsec@lists.tislabs.com) or to the editor.

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.

Abstract

This document describes how to detect one or more network address translation devices (NATs) between IPsec hosts, and how to negotiate the use of UDP encapsulation of IPsec packets through NAT boxes in Internet Key Exchange (IKE).

INTERNET-DRAFT 10 Feb 2004

Table of Contents

1. Introduction

This document is split in two parts. The first part describes what is needed in IKE Phase 1 for NAT-Traversal support. This includes detecting if the other end supports NAT-Traversal, and detecting if there is one or more NAT between the peers.

The second part describes how to negotiate the use of UDP encapsulated IPsec packets in IKE's Quick Mode. It also describes how to transmit the original source and destination addresses to the peer if required. The original source and destination addresses are used in transport mode 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 [Hutt03] describes the details of UDP encapsulation and [Aboba03] provides background information and motivation of NAT-Traversal in general. This document, in combination with [Hutt03] represents an "unconditionally compliant" solution to the requirements as defined by [Aboba03].

The basic scenario for this document is the case where the initiator is behind NA(P)T and the responder has a fixed static IP address.

This document defines a protocol that will work even if both ends are behind NAT, but the process of how to locate the other end is out of the scope of this document. In one scenario, the responder is behind a static host NAT (only one responder per IP as there is no way to use any other destination ports than 500/4500), i.e. it is known by the

---

configuration.

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 support for NAT-Traversal and detection of NAT along the path between the two IKE peers occurs in IKE [RFC-2409] Phase 1.

The NAT may change the IKE UDP source 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 change the source port:

o only one IPsec host behind the NAT

o for the first IPsec host the NAT can keep the port 500, and the NAT will only change the port number for later connections

Recipients MUST reply back to the source address from the packet (See [Aboba03] section 2.1, case d). This also means that when the original responder is doing rekeying, or sending notifications etc. to the original initiator it MUST send the packets using the same set of port and IP numbers that was used when the IKE SA was last time used.

For example, when the initiator sends a packet having source and destination port 500, the NAT may change that to a packet which has source port 12312 and destination port 500. The responder must be able to process the packet whose source port is that 12312. It must reply back with a packet whose source port is 500 and destination port 12312. The NAT will then translate this packet to have source port 500 and destination port 500.

3.1. Detecting support of Nat-Traversal

The NAT-Traversal capability of the remote host is determined by an exchange of vendor ID payloads. In the first two messages of Phase 1, the vendor id payload for this specification of NAT-Traversal (MD5 hash of "RFC XXXX" - ["XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX"]) MUST be sent if supported (and it MUST be received by both sides) for the NAT-Traversal probe to continue.

[Note to the RFC Editor: The XXXX is replaced with the RFC number of this document when the number is known. The XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX will be replaced with MD5 hash of the text "RFC XXXX" (the exact hex string will be provided by the authors when the rfc number is known). This instruction is to be removed from the final RFC].

3.2. Detecting the presence of NAT

The purpose of the NAT-D payload is twofold, It not only detects the presence of NAT between the 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.

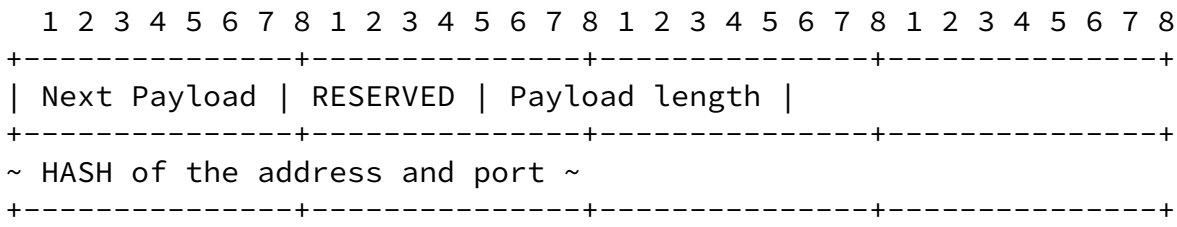
To detect NAT between the two hosts, we need to detect if the IP address or the port changes along the path. This is done by sending the hashes of the IP addresses and ports of both IKE peers from each end to the other. If both ends calculate those hashes and get same result they know there is no NAT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

If the sender of the packet does not know his own IP address (in case of multiple interfaces, and the implementation does not know which IP address is used to route the packet out), the sender can include multiple local hashes to the packet (as separate NAT-D payloads). In this case, 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 the normal case there are only two NAT-D payloads.

The NAT-D payloads are included in the third and fourth packet of Main Mode, and in second and third packet in the Aggressive Mode.

The format of the NAT-D packet is



The payload type for the NAT discovery payload is 15.

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 an IPv4 address and 16 octets for an IPv6 address. The port number is encoded as a 2 octet number in network byte-order. The first NAT-D payload contains the remote end's IP address and port (i.e. the destination address of the UDP packet). The remaining NAT-D payloads contain possible local end IP addresses and ports (i.e. all possible source addresses of the UDP packet).

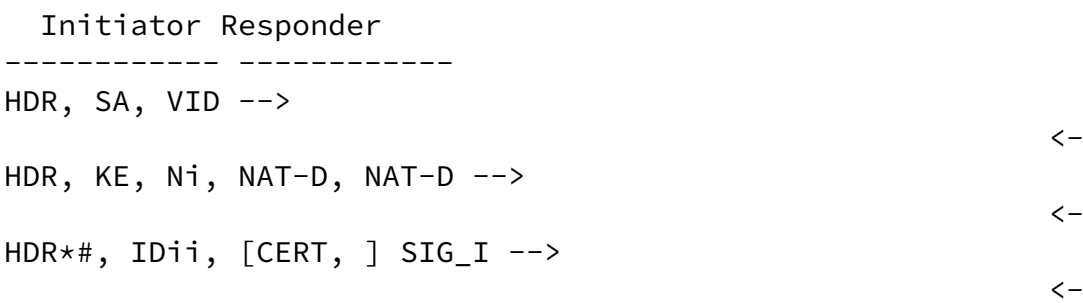
If there is no NAT between the peers, the first NAT-D payload received should match one of the local NAT-D payloads (i.e. the local NAT-D payloads this host is sending out), and one of the other NAT-D payloads must match the remote end's IP address and port. If the first check

---

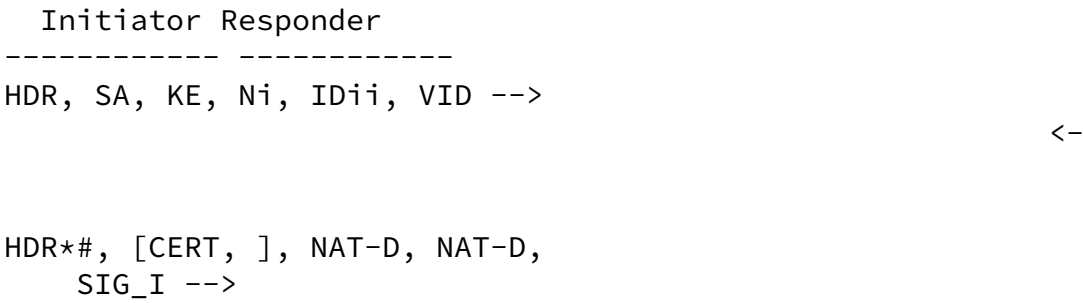
fails (i.e. first NAT-D payload does not match any of the local IP addresses and ports), then it means that there is dynamic NAT between the peers, and this end should start sending keepalives as defined in the [Hutt03] (this end is behind the NAT).

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

An example of a Phase 1 exchange using NAT-Traversal in Main Mode (authentication with signatures) is:



An example of Phase 1 exchange using NAT-Traversal in Aggressive Mode (authentication with signatures) is:



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

4. Changing to new ports

IPsec-aware NATs can cause problems (See [Aboba03] section 2.3). Some NATs will not change IKE source port 500 even if there are multiple clients behind the NAT (See [Aboba03] section 2.3, case n). They can also use IKE cookies to demultiplex traffic instead of using the source port (See [Aboba03] section 2.3, case m). 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.

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

In Main Mode, the initiator MUST change ports when sending 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 informational notifications) MUST be sent on port 4500. In addition, the IKE data MUST be prepended with a non-ESP marker allowing for demultiplexing of traffic as defined in [Hutt03].

Thus, the IKE packet now looks like:

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

assuming authentication using signatures. The 4 bytes of non-ESP marker is defined in the [Hutt03].

When the responder gets this packet, the usual decryption and processing of the various payloads is performed. If this is successful, the responder MUST update local state so that all subsequent packets (including informational notifications) to the peer use the new port, and possibly the 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-changed IP address. The responder MUST respond with all subsequent IKE packets to this peer using UDP(4500,Y).

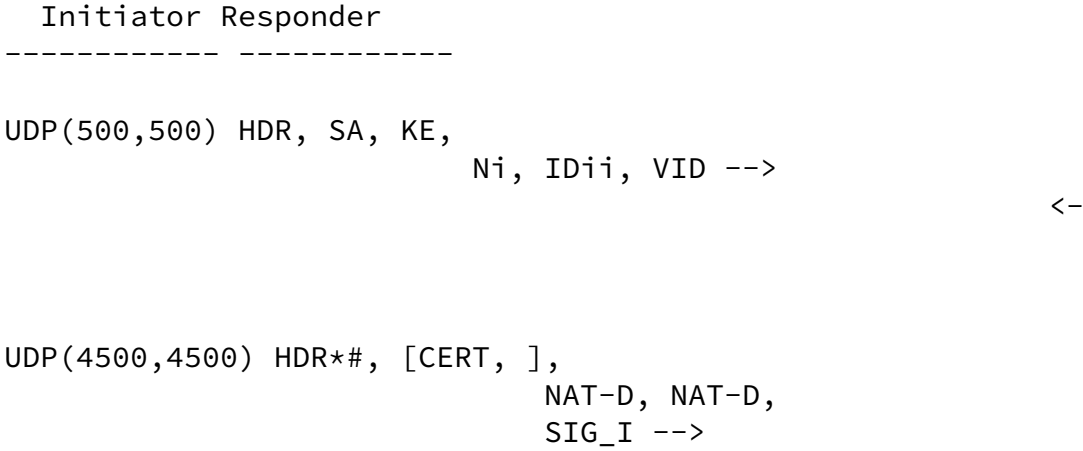
Similarly, if the responder needs to rekey the Phase 1 SA, then the rekey negotiation MUST be started using UDP(4500,Y). Any implementation that supports NAT traversal MUST support negotiations that begin on port 4500. If a negotiation starts on port 4500, then it doesn't need to change anywhere else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

Once port change has occurred, if a packet is received on port 500, that packet is old. If the packet is an informational packet, it MAY be processed if local policy allows. If the packet is a Main Mode or Aggressive Mode packet (with same cookies than previous packets), it SHOULD be discarded. If the packet is new Main Mode or Aggressive exchange then it is processed normally (the other end might have rebooted, and this is starting new exchange).

Here is an example of a Phase 1 exchange using NAT-Traversal in Main Mode (authentication with signatures) with changing port:

```
  Initiator Responder
------------ ------------
UDP(500,500) HDR, SA, VID -->
                                                    <-
UDP(500,500) HDR, KE, Ni,
                             NAT-D, NAT-D -->
                                                    <-
```

The procedure 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 it's internal IKE ports. The responder MUST respond with all subsequent IKE packets to this peer using UDP(4500,Y).



If the support of the NAT-Traversal is enabled the port in the ID payload in Main Mode/Aggressive Mode MUST be set to 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 changes 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 port change case involves out-of-band discovery of the ports to use. Those discovery methods are out of scope of this document. 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), it 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 change need take place. Also, the first keepalive timer starts after the change to the new port, no keepalives are sent to the port 500.

5. Quick Mode

After the Phase 1 both ends know if there is a NAT present between them. The final decision of using NAT-Traversal is left to Quick Mode. The use of NAT-Traversal is negotiated inside the SA payloads of Quick Mode. In Quick Mode, both ends can also send the original 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.

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 3
UDP-Encapsulated-Transport 4

It is not normally useful to propose both normal tunnel or transport mode and UDP-Encapsulated modes. UDP encapsulation is required to fix the inability to handle non-UDP/TCP traffic by NATs (See [Aboba03] section 2.2, case i).

If there is a NAT box between hosts, normal tunnel or transport encapsulations may not work and in that case UDP-Encapsulation SHOULD be used.

If there is no NAT box between, there is no point of wasting bandwidth by adding UDP encapsulation of packets, thus UDP-Encapsulation SHOULD NOT be used.

Also, the 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 and destination addresses

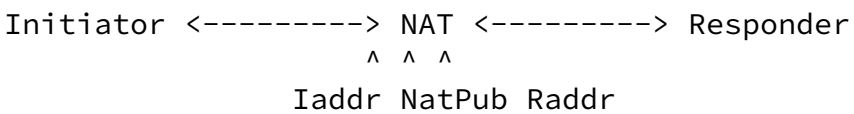
In order to perform incremental TCP checksum updates, both peers may

need to know the original IP addresses used by their peer when that peer constructed the packet (See [Aboba03] section 2.1, case b). For the initiator, the original Initiator address is defined to be the Initiator's IP address. The original Responder address is defined to be the perceived peer's IP address. For the responder, the original Initiator address is defined to be the perceived peer's address. The original Responder address is defined to be the Responder's IP address.

The original addresses are sent using NAT-OA (NAT Original Address) payloads.

The Initiator NAT-OA payload is first. The Responder NAT-OA payload is second.

Example 1:



The initiator is behind a NAT talking to the publicly available responder. Initiator and Responder have IP addresses Iaddr, and Raddr. NAT has public IP address NatPub.

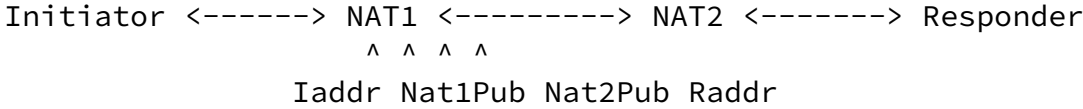
Initiator:

NAT-OAi = Iaddr
NAT-OAr = Raddr

Responder:

NAT-OAi = NATPub
NAT-OAr = Raddr

Example 2:



Here, NAT2 "publishes" Nat2Pub for Responder and forwards all traffic to that address to Responder.

Initiator:

NAT-OAi = Iaddr

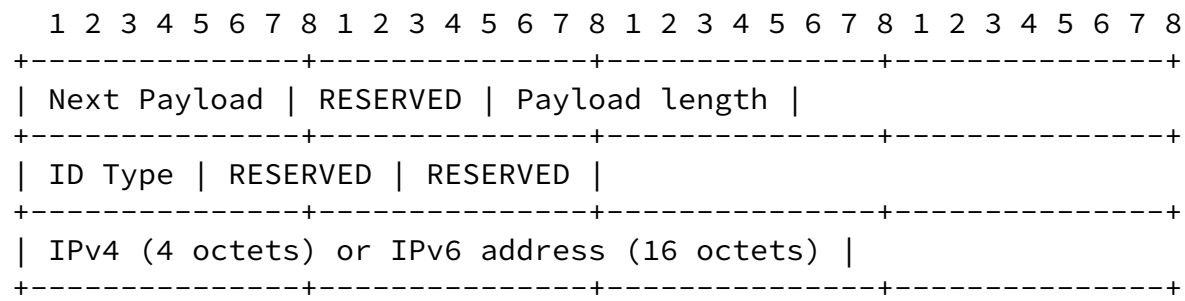
```
                    NAT-OAr = Nat2Pub

Responder:
                    NAT-OAi = Nat1Pub
                    NAT-OAr = Raddr
```

In case of transport mode both ends MUST send the both original Initiator and Responder addresses to the other end. For tunnel mode both ends SHOULD NOT send original addresses to the other end.

The NAT-OA payloads are sent inside the first and second packets of Quick Mode. The initiator MUST send the payloads if it proposes any UDP-Encapsulated-Transport mode and the responder MUST send the payload only if it selected UDP-Encapsulated-Transport mode, i.e. it is possible that the initiator sends the NAT-OA payload, but proposes both UDP-Encapsulated transport and tunnel mode. Then the responder selects the UDP-Encapsulated tunnel mode and does not send the NAT-OA payload back.

The format of the NAT-OA packet is



The payload type for the NAT original address payload is 16.

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-OAi, NAT-OAr] -->
                                                                    <-
```

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 (NAT can change them), so the IP and port numbers MUST NOT be used for determining which IKE/IPsec SAs to remove (See [Aboba03] section 2.1, case c). The ID payload sent from the other end SHOULD be used instead, i.e. when an 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 this, ends which are NOT behind NAT SHOULD use the last valid authenticated packet from the other end to determine which IP and port addresses should be used. The host behind dynamic NAT MUST NOT do this as otherwise it opens a DoS attack possibility, and there is no need for that, because the IP address or port of the 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 detect 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 probes reveal NAT-Traversal support to anyone watching the traffic. Disclosure that NAT-Traversal is supported does not introduce new vulnerabilities.

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, authentication measures other than IP addresses should be used). This means that authentication using pre-shared-keys cannot be used in Main Mode without using group

shared keys for everybody behind the NAT box. Using group shared keys is huge risk because it allows anyone in the group to authenticate to any other party and claim to be anybody in the group, i.e. a normal user could be impersonating a vpn-gateway, and acting as a man in the middle, and read/modify all traffic to/from others in the group. 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 an educated guess of the private address space is done, then the number of hash calculations needed to find out the internal IP address goes down to 2^24 + 2 * (2^16).

o Neither NAT-D payloads or Vendor ID payloads are authenticated at all in Main Mode nor in 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 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.

o Updating the IKE SA / ESP UDP encapsulation IP addresses and ports for each valid authenticated packet can cause DoS in the case where we have an attacker who can listen to all traffic in the network, and can change the order of the packets and inject new packets before the packet he has already seen, i.e. the attacker can take an 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 the 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 SHOULD AUDIT the event every time the mapping is changed, as in the normal case it should not happen that often.

9. IANA Considerations

This documents contains two new "magic numbers" which are allocated from

the existing IANA registry for IPsec. This document also renames existing registered port 4500. This document also defines 2 new payload types for IKE, and there is no registry for those in the IANA.

New items to be added in the "Internet Security Association and Key

---

Management Protocol (ISAKMP) Identifiers" Encapsulation Mode registry:

Name	Value	Reference
UDP-Encapsulated-Tunnel	3	[RFC XXXX]
UDP-Encapsulated-Transport	4	[RFC XXXX]

Change in the registered port registry:

Keyword	Decimal	Description	Reference
ipsec-nat-t	4500/tcp	IPsec NAT-Traversal	[RFC XXXX]
ipsec-nat-t	4500/udp	IPsec NAT-Traversal	[RFC XXXX]

New IKE payload numbers are (There is no IANA registry related to this, and no need to create new one, but if one is added these should be added to there):

NAT-D	15	NAT Discovery Payload
NAT-OA	16	NAT Original Address Payload

10. Intellectual property rights

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the IETF's procedures with respect to rights in IETF Documents can be found in RFC XX and RFC XY. [note to RFC Editor - replace XX with the number of IETF IPR and replace XY with number of IETF SUB.]

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at

http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

11. 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 document used as the base for this document.

---

12. Normative 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

[Hutt03] Huttunen, A. et. al., "UDP Encapsulation of IPsec Packets", draft-ietf-ipsec-udp-encaps-06.txt, January 2003

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

[IETF SUB] Bradner, S., "IETF Rights in Contributions", draft-ietf-ipr-submission-rights-08.txt, October 2003

[IETF IPR] Bradner, S., "Intellectual Property Rights in IETF Technology", draft-ietf-ipr-technology-rights-12.txt, October 2003

13. Non-Normative References

[Aboba03] Aboba, B. et. al., "IPsec-NAT Compatibility Requirements", draft-ietf-ipsec-nat-reqts-06.txt, October 2003.

14. Authors' Addresses

Tero Kivinen

SafeNet, Inc.
Fredrikinkatu 47
FIN-00100 HELSINKI
Finland
E-mail: kivinen@safenet-inc.com

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

---

15. Full copyright statement

Copyright (C) The Internet Society (year). This document is subject to the rights, licenses and restrictions contained in RFC XXXX and except as set forth therein, the authors retain all their rights.

[Note to the RFC Editor - XXXX above to be replaced with the number of [IETF SUB]]

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/S HE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.