IPsec over NAT Justification for UDP Encapsulation

draft-ietf-ipsec-udp-encaps-justification-00.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.

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

1. Abstract

This draft explains the design justification and alternatives for two IPsec over NAT drafts, UDP Encapsulation of IPsec Packets, [Hutt01] and [Kiv00]. This draft sets the requirements for a solution in terms of scenarios in which NAT is used, and NAT operations. This draft specifies the requirements for IPsec NAT traversal, scenarios these extensions enable, and design rationale for the proposed solution. This draft assumes that the reader is familiar with the interactions of NAT with IPsec documented in [Aboba04].

2. Conventions used in this document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. A packet which is processed through a network address translator (NAT) or a network address and port translator (NAPT) is said to be "NATed".
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119]

3. Introduction

NAT is widely used for Internet "connection sharing" and ships as a standard feature in every router and edge device. Cable and DSL deployments have been known to install NAT in embedded devices resident inside the home. Additionally most hotel Internet connections are using NAT. Many airport & wireless services in public areas use NAT to connect to the Internet. Some Internet Service Providers use a centralized NAT to connect their clients to the Internet.

Without a solution to the problem, the L2TP/IPsec VPN client and an IPsec tunnel mode VPN client can not connect from home or from business sites which use NAT to connect to the Internet. Likewise, transport TCP and UDP traffic that is protected by IPsec transport mode can not be exchanged between peer to peer or server-to-server applications.

Without an IETF standard solution for IPsec over NAT, vendor products will not interoperate, and will be forced to support multiple methods.

This proposal allows ESP to pass through NAT if the NAT allows UDP traffic. It does not require the NAT to be aware of IPsec and IKE negotiations. The technique is used only if the initiator and the responder support it, and only used when necessary, since NAT detection is built into the protocol.

This technique allows L2TP protected by IPsec transport mode using the most popular and default encryption format (IPsec ESP) to go through existing NATs - which do either port and address translation of UDP packets or pure-address translation.

It also allows IPsec tunnel mode to go through the same type of NATs, a requirement for many in the industry which implement extensions to IPsec tunnel mode for remote access. Address assignment can also use the standards track DHCP over IPsec method.

Note however that when RFC IPsec tunneling through a NAT between an end system (static internal & external IP) and a gateway, or between two gateways - the internal addresses used by packets inside the IPsec tunnel can not be changed by the NAT. So the internal IP packet will retain a potentially invalid address on the destination network. There are three options to provide proper addressing for tunneled packets:

- perform a second NAT function either before the tunnel ingress or after the tunnel egress.
- In the case where traffic inside the tunnel is clear text TCP &
UDP traffic, this is normal NAT and not affected by the IPsec tunnel being NATed itself.

- In the case where traffic inside the IPsec tunnel is IPsec transport traffic, this spec provides the mechanism for the IPsec transport peers to detect this second NAT and pass it accordingly.

- Not use the RFC method of negotiating an IPsec tunnel - instead use proprietary extensions to RFC 2409 for address assignment for remote access VPN tunnels.

- Keep the internal address as is, and configure the destination network to be able to handle it.

This method should be able to be used in all scales where NAT is deployed today to do simple pure address-to-address, or address and port translation. However, it will not be able to be used where the NAT would otherwise perform packet inspection and replacement of address or port information inside the packet (e.g. modifying the FTP PORT command packet). Thus UDP encapsulated IPsec transport and tunnel traffic will be successful if the communications model is that the client initiates TCP connections & UDP sessions from itself out through the NAT.

This protocol does not involve the client communicating directly with the NAT for the purpose of opening ports to receive inbound connections. If required, another method such as RSIP may be used by the application first to communicate with the NAT.

IKE & IPsec issues with NAT are documented in [Aboba04]. This proposal provides a solution to all issues identified when using IKE Identity Protect Mode (Main Mode) or Aggressive Mode and Quick Mode. Other proprietary IKE extensions may be present, but are not considered by this draft. Address assignment for a remote access VPN IPsec tunnel client can use DHCPv4 Configuration of IPsec Tunnel Mode [Patel12].

Most importantly, this proposal does not require change to the NAT device itself - which is seen as practically too difficult to accomplish. The deployment of IPsec enabled systems is typically controlled by the user of that system, and thus can be updated independently of the network infrastructure at ISPs, hotels, mobile and temporary networks, and other places where NAT is required to overcome limited IPv4 Internet routeable address availability. We expect the technique described here to be used in the interim to pass IPv4 addressed traffic only. We expect this technique to become outdated as IPv6 is deployed, enabling end-to-end IPsec without address translation.

4. <Section 4 heading as appropriate>

Dixon, et. al.

IPsec over NAT Justification for UDP Encapsulation June 2001

Aboba presents requirements for an IPsec over NAT solution. The [Kiv00]and [Hutt01] drafts do this as follows:

1. Deployability - the design ensures that IKE and IPsec ESP
appear only as UDP protocol packets which are allowed to have their source and destination addresses changed by a common NAT. An IPsec aware NAT is not required. Thus the network infrastructure need not change. Firewalls can filter on a single UDP port. No change in DNS clients, DNS servers, DHCP clients, DHCP servers, or applications programs are required for this IPsec over NAT solution, as is typically required for IPv6 deployment. There is no communication between host and NAT device. The timeframe for deployment is simply the timeframe for end systems to have a software patch or update which enables the protocol functionality defined here.

2. Telecommuter scenario. While this draft claims no need for support beyond that needed for telecommuter scenarios, the IPsec protocol is more generally useful than for VPN remote access. Thus our design solves remote access scenarios and makes general IPsec transport mode or tunnel mode work through single and multiple NAT devices.

3. Scaling. The solution specified our drafts will provide for scaling on the receiving end on the order which memory, CPU and hardware acceleration allow for RFC IPsec. There is no inherent limitation in our design to scaling. Overlapping SPD entries are addressed by the treatment of values contained in the main mode and aggressive mode ID payloads, and Proxy ID payloads.

4. Mode support. Our design provides for both IKE Identity Protect Mode and Aggressive Mode to negotiate IPsec transport and tunnel mode UDP ESP encapsulation. Our wire format and IKE design do specify a solution for AH encapsulation through NAT. However, this capability is included only so the drafts as standards track documents offer a "complete solution". Practically, we do not expect an implementation to support AH over NAT, preferring ESP null encryption with integrity, or an IPsec ESP tunnel mode for full packet encapsulation instead.

5. Interoperability. Backward compatibility is assured through the continued use of UDP 500 with additional payloads to announce the capability of NAT discovery and traversal. To the extent that existing implementations properly discard payloads they do not understand, an IPsec-over-NAT capable client will interoperate cleanly with an earlier IPsec/IKE client without the presence of NAT.

6. Security. The security considerations section of [Kiv00] discusses specifically the security implications to IKE of including new payloads. It was decided to leave these payloads unauthenticated in order to more expediently reach a solution for IPsec over NAT, rather than adopt new proposals for the authentication of all payloads in IKE phase 1 and phase 2 discussed by the now expired [Kiv-HASH-02]. Further, we specifically did not change anything about verifying the contents of a tunnel match the address selectors negotiated in the Proxy ID. And we provide a NAT-OA original address payload so that the authenticated IPsec receiving peer may either reconstruct the original packet headers or take some other action based on knowledge of the original address...
for the purpose of anti-spoofing checks during packet processing.

Additionally the following goals were set by the authors for an IPsec over NAT solution:
1. One IETF standardized way for IKE/IPsec NAT traversal.
2. Minimize the use of UDP encapsulation overhead
   a. Discover if NAT is present in the path between the two IKE peers, and discover which is behind the NAT.
   b. Allow for products to be configured to not perform UDP encapsulation when the user knows the presence of the NAT, if its IPsec-friendly capability. Product works through IPsec-friendly NATs.
3. Transport IKE and IPsec traffic through the NAT without forcing the NATs to be modified.
4. Keep the NAT mappings (or state firewall mappings) alive for the duration of the IKE and IPsec sessions.
5. Keep IKE as simple as possible for ease of interop, better security.
6. Minimize keep-alive traffic to reduce the number of timers required.

5. NAT Scenarios

This section identifies common network topologies where NAT is used to pass traffic between address boundaries. A NAT device may also be viewed as a stateful filtering device that performs no translation of the address or ports in the packet, but creates mapping states specific to the address and port information in a packet.

All below scenarios may support an IPsec transport/tunnel SA for all possible traffic, where 1 or more TCP or UDP or other protocol sessions may be protected by the single IPsec transport/tunnel security association. Security associations are assumed to exist in pairs. There may be multiple transport/tunnel SAs of finer granularity as well.

The term "private IP address" used here is not restricted to be the IANA defined private IP address space, but rather is exemplary of an address used within an address boundary defined by the NAT device.

A. One or more clients behind a NAT connecting simultaneously to a server's Internet IP address.

C1 private IP <-> NAT Internet IP <----> Internet IP S1

NAT is doing pure address translation 1:1, or doing many:1 address translation with port translation.
B. One or more clients behind NAT1 connecting simultaneously to the same dest IP address which is behind NAT2

C1 private IP <-> NAT1 Internet IP <-> Internet IP NAT2 <-> private IP S1

NAT1 is performing many:1 port address translation (onto Internet for example), NAT2 is performing 1:1 address translation (into DMZ for example)

C. One or more clients through 2 or more outbound NATs to destination Internet IP address

C1 private IP <-> NAT1 private IP <-> NAT2 Internet IP <-> Internet IP S1

NAT1, NAT2 are performing many:1 port address translation.

D. Internet client connecting to server behind NAT.

C1 Internet IP <-> Internet IP NAT1 <-> private IP S1

NAT would most likely be doing 1:1 address translation, but may be publishing an address and port combination that is fully translated to an private address and different port.

E. Multiple clients identically configured (i.e. C1 and C2 have the same private network IP address assigned to them, e.g. 192.168.0.2), each behind their own NAT, connected to same Internet IP of Server1.

C1 private IP <-> NAT1 Internet IP <-> Internet IP S1
C2 private IP <-> NAT2 Internet IP <-> Internet IP S1

F. Gateway to gateway tunnel over NAT.

C1 private IP <-> GW1 <-> NAT1 Internet IP <-> GW2 <-> private IP S1

An IPsec tunnel SA is established between GW1 and GW2. NAT1 performs address translation on the tunnel packets.

Dixon, et. al. IPsec over NAT Justification for UDP Encapsulation June 2001

6. Design Overview and Rationale>

The IETF standard for NAT traversal by IPsec should be unencumbered from intellectual property claims to ensure rapid adoption by the IPsec vendor community.

We propose to UDP encapsulate the ESP traffic, thereby allowing it through the NAT.

During design several different criteria needed to be balanced - Encapsulation using UDP port 500 was chosen so as to enable easy deployment without the need to change firewall rules used by corporations.
The chosen method UDP encapsulates ESP traffic as efficiently as possible. AH and future protocols may be UDP encapsulated, but at greater expense. This ensures that the most widely needed traffic protection method is most efficient.

Encapsulation using UDP port 500 makes each packet 8 bytes larger than encapsulation using a different port, this is however balanced by the reduced need to have keepalive packets, and reduced complexity in IKE.

7.0 UDP-encapsulated ESP Header Format

7.1 Not using TCP

UDP header provided minimal standard encapsulation, 8 bytes, that would traverse exiting NATs. TCP would incur a minimal 20byte header.

NAT and firewall devices are aware of TCP connection setup traffic pattern. Thus these devices expect to see a real TCP connection establish and operation, but to varying degrees. It was a non-goal of this work to enable IPsec work through a TCP proxy. Thus a TCP encapsulation would only be for the purpose of exposing a useless header for NAT traversal.

We could run IKE over TCP or we could use IKE as it is and just use TCP for IPsec SA packet encapsulation. A real TCP connection would degrade the reliability of communication which the security features of IPsec SAs and packet format provide. For both IKE and IPsec, the traffic flow channel is not subject to disruption, except by packet loss or packet modification. Using a real TCP connection to transport either IKE or IPsec would introduce vulnerability to trivial TCP connection reset attacks by spoofed TCP packets. Thus logic would have to be implemented to constantly re-establish the TCP connection whenever it went down. Which side would do this? It would have to be the side that originally established the communication if stateful filtering were in place. Therefore a real TCP connection would not be desired. Rather a fake TCP handshake and then using a TCP header without sequence number tracking, ACKing, etc. An additional weakening of the security properties of IKE would be introduced if TCP FIN or RST flags where used in combination with IKE SA delete messages. Thus a keep-alive mechanism would need to be invented for this partial TCP functionality.

There is some evidence to suggest that when IPsec transported a normal TCP connection over yet another full TCP connection, the two TCPs will interact with each other to slow down data transfer [Titz].

Routers handle TCP so to ensure in order delivery of TCP segments. Most TCP implementations require nor more than 3-5 packets received out of order. There is no such requirement for UDP, only the IPsec SA anti-replay window.

It was the consensus of the authors that the complexity of faking
TCP and the additional overhead of the TCP header made UDP the preferred choice for encapsulation.

7.2 Unifying Control and Data traffic on UDP 500

A single UDP src, dst port mapping is used for both IKE and IPsec packets. Since the IKE initiation is sent to the well-known ISAKMP destination port 500, the receiver must use the UDP source port as the packet is received (inbound SA) to be the destination port of the return traffic (outbound SA). The source port of the return traffic must be the destination port observed when the inbound SA packet was received. This is to ensure that NAT port mappings are properly preserved.

This draft uses the well-known IKE UDP port 500 to encapsulate both IKE control traffic and ESP data traffic. This approach allows for:

- a single well known port over which ESP UDP encapsulation would be done. The IKE port is already assigned, and well-known for IPsec so existing documentation and training won’t have to be changed.
- It is easiest to explain how to allow IPsec traffic, simply open UDP port 500
- A single approach allows firewall configurations to ship with defaults
- minimization of keep-alive/heartbeats for both IKE & ESP — as long as there was traffic flowing middle devices like stateful FWs & NATs would keep the mapping alive

8. Alternatives To Unified Data & Control Considered

In the following, "501" refers to a newly defined (eventually by IANA) well-known port for IKE. Using "501" either for IKE and/or for the UDP encapsulation of the IPsec packets would allow us to change the header formats of IKE and potentially squeeze out more efficiency for the more common IPsec traffic. For example, the format could be:

\[IP][UDP 1025, 501][IPsec ESP traffic]\n
And

\[IP] [UDP 1025, 501][<4 bytes 0>][<4 byte TYPE>][IKE or ESP traffic.]

In the second case, the <4 bytes 0> would line up with the ESP header’s SPI, which would never be 0, therefore distinguishing the difference if both were encapsulated on 501. This clearly has the benefit of being tighter on the wire for the more common ESP traffic. The alternatives below will show why we discarded these (and similar) encapsulation schemes.

8.1 Using a non-500 dest port for "IPsec over NAT" negotiation traffic

One desired benefit of using 501 was to completely isolate the
IKE/IPsec over NAT from the standard IKE/IPsec implementation. We would still keep IKE and ESP on the same port pair, and thus essentially defined a new well known port and protocol for it, preserving UDP 500 as "unpolluted" for RFC 2409 use only. However, this does not meet [Aboba04] requirement of backward compatibility. Also, UDP 500 is already extended in many ways with special behavior indicated through the use of vendor-id payloads.

The operational issue is how the initiator knows to use this UDP dest port 501 for the first packet when IKE initiates. It could be configured if the initiator knows it is behind a NAT. However this seems very impractical as NATs are generally invisible to the end system behind it and there would be no way to establish trust with the hotel NAT, the home NAT, the ISP NAT, etc.

The deciding factor is that we only want to use the UDP encapsulation when necessary, that is when a NAT is discovered. So we conclude that NAT discovery must be part of the initial IKE negotiation for this technique to be widely deployable.

The initiator could attempt 500 and then retry on 501 at the heavy expense of a timeout. Or the initiator could send simultaneously to 501 and to 500. The 501 packet could include NAT discovery, while the 500 packet would not. Either we receive no reply to the 500 packet, or we do but also receive a reply from 501 indicating NAT. Clearly timing may introduce a mistake in the decision of which mode to use. Better to avoid this complexity altogether and work within the standard IKE message exchange state machine.

To know that 500 did not succeed because of a NAT, we’d need discovery packets in the port 500 usage negotiation. To reply back to the initiator that NAT was discovered, the port 500 receiver would have to reply back to the as-received source port. For some implementations of IKE, this is also a change. So we saw several changes required in IKE using port 500 to enable usage of 501. It was then clear that NAT discovery using port 500 was the way to go. A new NAT-traversal behavior would be engaged by first the capability message, and secondly by the actual discovery payloads.

8.2 Initial contact on 500, discovery, move IKE to well known port 501

We could still move the negotiation off of 500 when NAT was discovered. In the middle of the negotiation, it is possible for both peers to float to using 501 at the same time. This would be for the eventual purpose of using the same src, dst 501 port pair (NAT mapping) for the IPsec SAs bound to the IKE exchange to aggregate keep-alives. The initial outbound mapping for 500 would expire on the NAT, followed by a new mapping would be established by the completion of IKE and maintained by the IPsec SA packet flow.

The main problem here is when and how to float. This would have to be clearly defined with MUST statements to assure interoperability. Clearly, the easiest time to float would be at the end of Main Mode, but before Quick Mode started. However, there is the design requirement that we allow for negotiation of encapsulation type in
Quick Mode to support the case of allowing IPsec-friendly NATs to pass non-UDP encapsulated ESP traffic. So, if we floated to 501 before Quick Mode, we will do that unnecessarily in the IPsec-friendly NAT case. If we delay floating to 501 until after both sides have agreed to do UDP encapsulation in the quick mode SA payload exchange, then we have timing conditions since there may be multiple quick modes going, and floating any one quick mode, floats the main mode under it, to the potential detriment of the other quick modes.

Rekey introduces additional interop complexity. Assume the float to 501 by the above means happens at the end of main mode. Assume the peer outside the NAT is rekeying the Main Mode. Since IKE floated to 501, there are no keep alives keeping a UDP 500 hole open, so the peer will need to initiate the MM to 501 instead of 500. Now we have to handle the case where the main mode starts on 501 as well.

Finally, we consider the firewall configuration. The case where neither side has initiated, but either side could, means the firewall configuration would have to allow inbound requests on both sides destined to 500 and to 501. The authors didn’t think this was a major reason, but one to mention as we have all experienced resistance to opening ports from customer firewall administrators.

8.3 Initial contact on 500, discovery, move IKE to dynamically chosen port

In addition to all the complexities above, this approach makes static port filtering in effective. Handling the idle (no active IPsec quick mode) and rekey cases make this the most difficult choice with which to achieve interoperability. The typical scenario of double firewalls would pose the greatest difficulty, as IKE can initiate from either side.

8.4 Initial contact and maintain IKE on 500, use 501 for IPsec only

The motivation for this approach is to avoid an 8 byte cookie=0x00 as the demultiplexing value to separate ESP traffic from IKE traffic which is used when all traffic uses only port 500.

This approach means that we must keep both the 500 and 501 NAT mappings alive. The only real problem with this approach is that is depends on the initiator of the negotiation to also be the initiator of the UDP src xxx, dst 501 packet in order to establish the NAT mapping. This is not always the case.

In scenario A, where the client initiated from behind the NAT to establish the NATI translation to UDP source 1025, dest 500, then sent another outbound UDP packet to 501, which the NAT would translate to UDP src 1026, dest 501 - this would work initially. Assume then that the client performs a long running download of data, causing the gateway to rekey quick mode first. When the gateway, still using the UDP 1025, 500 mapping for IKE, completed
the quick mode, it could not send to UDP destination 501 because the client’s NAT does not have a mapping for this. Instead, the gateway must continue to use UDP 1026, 501 for the UDP encapsulation of the new ESP SA.

Now consider that the session goes idle – idle from the point of view of the IPsec SA being used to send protected packets. The client is still sending keep alive packets to maintain the client mapping on the NAT. But the gateway doesn’t necessarily get these. So the gateway can delete the IPsec SA, as can the client. Both may or may not make the decision to delete their inbound or outbound SA at any time. The notify messages may or may not be sent and may or may not be honored by the receiver. Perhaps an outbound packet arrives to the IPsec system which would use the already deleted IPsec SA. The gateway rekeys to the client using quick mode on UDP 1025, 500. The gateway now must choose a UDP source port with dest 501 for the UDP encapsulation of the new IPsec SA. The client’s NAT won’t have a mapping for anything other than what the client used previously. So the gateway MUST use the previous mapping, and the client MUST keep the mapping alive for deleted IPsec SAs. The client is not the sender of the first ESP packet on the new quick mode. This could be fixed by the client "punching out" with a keep-alive message, in anticipation of the incoming UDP ESP packet from the gateway destined for UDP 501 on the client. But the complexity and timing conditions can be avoiding by always using the established IKE UDP mapping. Interoperability is also better served by keeping the same mapping.

Using both 501 and 500 doubles the keep-alives since both must be kept current on the NAT, which increases the state required for each IPsec SA. And again, using both 500 and 501 requires firewall administrators to open both for bidirectional communication.

9. Addresses in IKE ID Payloads

9.1 IKE Main Mode ID Payload

If a NAT is present, certain limitations are placed upon the valid IDs exchanges in IKE. RFC 2408 and 2407 specify the ID payload to represent an identity of the initiator for the purpose of the responder using it for a policy lookup. Section 4.6.2.1 Identification Type Values allows many types. However in practice, implementations choose one or a few type values to support in their policy system. Backward compatibility is not a concern, as both initiator and responder must change to support this draft. However, backward compatibility with policy configuration systems is desired. The authors think that the IKE Main Mode ID payload value is largely being ignored or is not relevant to security policy lookup in the majority of IKE implementations, based on interop test results. We think this is because the receiver can not depend on an initiator (other than it’s own implementation) using a particular type value.

So [Kiv00] simply omits mention of the Main Mode ID payload, leaving it as a local matter. If the ID payload is meaningful in the implementation as an IP address, then that implementation must
decide how to handle the case where the initiator’s IP address put into the ID payload is invalid to the receiver. The receiver will know that NAT is present based on NAT discovery.

9.2 Addresses in IKE Quick Mode Proxy ID Payload

There was no consensus among the draft authors about particular language, so [Kiv00] does not address this at all. We note here that the IPsec packet processing comparison against the proxy id selector is a MUST via RFC 2401, 5.2.1 page 32, "In general, a packet’s source address MUST match the SA selector value."

This is most security critical in tunnel mode. But actually the tunnel mode proxy ID selectors would be the same as without NAT. So it's not an issue in tunnel mode. The tunnel mode Proxy ID using address selectors MUST remain a descriptor for the contents of the tunnel.

In transport mode ESP through NAT - to be consistent on this RFC 2401 MUST point, there are two ways to go:
1. the IKE-NAT-T initiator MUST propose 0.0.0.0 as source IP in the Proxy ID quick mode selector

2. the [Kiv00] responder MUST use the actual source IP of the packet to replace the source IP of the initiator’s selector before doing responder policy lookup with that selector.

#2 seems the best way to go to allow policy to be set for Internet source IPs (e.g. all traffic from DMZ Corp A would have source IP=xxx.xx.xx.xx when going outbound through their NAT - the selector at Corp B could be specific about traffic from Corp A.).

We are open to comments on an [Kiv00] section 4.3, something like:

"For transport quick mode, the proxy ID source IP address is sent by the initiator per RFC 2409. However the [Kiv00] compliant responder MUST use the actual source IP of the packet to replace the source IP of the initiator’s selector before doing responder policy lookup with that selector. This is a MUST, not SHOULD, to be consistent with RFC 2401 section 5.2.1 which requires selector comparison after decryption."

However, for transport mode, since the packet is authenticated by the successful processing of the hash, when the selector used in the Proxy ID was NOT an IP address, perhaps an FQDN, there is no meaning to the RFC 2401 MUST check of the packet against the selector.

So we have omitted specifying the Proxy ID payload contents and handling, leaving the processing as a local matter.

10. Keep-alive Mechanism

The keep-alive packet is used to maintain a viable UDP path between IPsec peers. It provides traffic flow in the absence of normal IKE
or IPsec traffic so that NAT and stateful filtering mappings remain active. The mechanism must work to maintain the mapping for the majority of existing NATs, without changing the NAT. The keep-alive packet is used strictly for path maintenance. It has no security function and thus does not use the cryptographic context of either IKE or IPsec. Thus a minimal UDP packet is sufficient for this purpose. Since it is not known how far on the path between peers the NAT occurs, the packet is sent with normal TTL values, not 1 or 2. The keep-alive SHOULD only be sent by the peer(s) behind the NAT to minimize keep alive traffic. The keep-alive SHOULD be silently discarded by the IPsec packet processor.

An informal survey of NAT mapping timeouts showed a minimum of 30 seconds as the time at which the mapping expired for UDP. There are IETF IPsec Working Group proposals on a keep-alive mechanism for IPsec or IKE peers, which uses the security context established between them. However, that keep-alive mechanism may not reach consensus in the working group before the NAT traversal work. That mechanism may be necessarily indistinguishable from normal IKE and IPsec traffic, and thus not filterable by the network to conserve bandwidth. Also, that mechanism need not be frequent enough to maintain a UDP path, since its purpose is to reliably verify that a peer cryptographic context exists (e.g. An IPsec tunnel still exists).

11. Security Considerations

The authors do not think this method exposes any security vulnerabilities into otherwise sound IPsec implementation. However, the following issues need to be addressed by implementers.

When using IPsec tunnel mode, clients connecting to the gateway may be using the same IP address in the inner IP header. One scenario where this is likely to happen is when the clients are using in the inner IP header their local private network address, and they are in different private networks. The GW MUST ensure that traffic destined towards the clients is sent only to the correct recipient.

When using IPsec tunnel mode, a client connecting to a gateway may be trying to steal an IP address from the network protected by the gateway. The QM initiated by the client MUST only be allowed by the gateway if:
- The QM IP address identities match the static policy defined at the gateway.
- The QM IP source address matches the address assigned to the client by the gateway.
- The gateway is configured to do further NAT to the packets, ensuring that whatever IP address the client is using, this cannot be used to steal local network IP addresses.
Any of these ensures that attacker is not possible to steal local network IP addresses.

An active attacker can change the UDP header such that the UDP destination port will be different that 500 or 501. This will cause a stream of IKE or ESP packets to be delivered to another port. An application listening on that UDP port may get flooded with packets
it can not understand. Denial of service for that other UDP application may be accomplished. Denial of service on the system may be accomplished if the processing of the bogus packets consumes CPU or crashes that service.

12. 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.

Dixon, et. al. IPsec over NAT Justification for UDP Encapsulation June 2001

13. References


14. Acknowledgements

The authors wish to acknowledge and thank Bernard Aboba, Cheryl Madson and Jonathan Trostle.

15. Author’s 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

Dixon, et. al.  
IPsec over NAT Justification for UDP Encapsulation June 2001

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  
XXX address missing  
lidiburro@nortelnetworks.com
Full Copyright Statement

"Copyright (C) The Internet Society (date). 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.