

IPsecME Working Group
Internet-Draft
Intended status: Informational
Expires: January 11, 2013

S. Hanna
Juniper
V. Manral
HP
July 10, 2012

Auto Discovery VPN Problem Statement and Requirements
draft-ietf-ipsecme-p2p-vpn-problem-02

Abstract

This document describes the problem of enabling a large number of systems to communicate directly using IPsec to protect the traffic between them. It then expands on the requirements, for such a solution.

Manual configuration of all possible tunnels is too cumbersome in many such cases. In other cases the IP address of end points change or the end points may be behind NAT gateways, making static configuration impossible. The Auto Discovery VPN solution is chartered to address these requirements.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 11, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	3
1.2. Conventions Used in This Document	4
2. Use Cases	5
2.1. Endpoint-to-Endpoint P2P VPN Use Case	5
2.2. Gateway-to-Gateway AD VPN Use Case	5
2.3. Endpoint-to-Gateway AD VPN Use Case	6
3. Inadequacy of Existing Solutions	7
3.1. Exhaustive Configuration	7
3.2. Star Topology	7
3.3. Proprietary Approaches	8
4. Requirements	9
4.1. Gateway and End Point Requirements	9
5. Security Considerations	10
6. IANA Considerations	11
7. Acknowledgements	12
8. Normative References	13
Authors' Addresses	14

1. Introduction

IPsec [RFC4301] is used in several different cases, including tunnel-mode site-to-site VPNs and Remote Access VPNs. Host to host communication employing transport mode also exists, but is far less commonly deployed.

The subject of this document is the problem presented by large scale deployments of IPsec and the requirements on a solution to address the problem. These may be a large collection of VPN gateways connecting various sites, a large number of remote endpoints connecting to a number of gateways or to each other, or a mix of the two. The gateways and endpoints may belong to a single administrative domain or several domains with a trust relationship.

Section 4.4 of RFC 4301 describes the major IPsec databases needed for IPsec processing. It requires an extensive configuration for each tunnel, so manually configuring a system of many gateways and endpoints becomes infeasible and inflexible.

The difficulty is that all the configuration mentioned in RFC 4301 is not superfluous. IKE implementations need to know the identity and credentials of all possible peer systems, as well as the addresses of hosts and/or networks behind them. A simplified mechanism for dynamically establishing point-to-point tunnels is needed. Section 2 contains several use cases that motivate this effort.

1.1. Terminology

Endpoint - A device that implements IPsec for its own traffic but does not act as a gateway.

Gateway - A network device that implements IPsec to protect traffic flowing through the device.

Point-to-Point - Direct communication between two parties without active participation (e.g. encryption or decryption) by any other parties.

Hub - The central point in a star topology, generally implemented in a gateway

Spoke - The edge devices in a star topology, implemented in endpoints or gateways

Security Association (SA) - Defined in [RFC4301].

1.2. Conventions Used in This Document

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 [RFC2119].

2. Use Cases

This section presents the key use cases for large-scale point-to-point VPN.

In all of these use cases, the participants (endpoints and gateways) may be from a single organization or from multiple organizations with an established trust relationship. When multiple organizations are involved, products from multiple vendors are employed so open standards are needed to provide interoperability. Establishing communications between participants with no established trust relationship is out of scope for this effort.

2.1. Endpoint-to-Endpoint P2P VPN Use Case

Two endpoints wish to communicate securely via a direct, point-to-point SA.

The need for secure endpoint to endpoint communications is often driven by a need to employ high-bandwidth, low latency local connectivity instead of using slow, expensive links to remote gateways. For example, two users in close proximity may wish to place a direct, secure video or voice call without needing to send the call through remote gateways, which would add latency to the call, consume precious remote bandwidth, and increase overall costs. Such a usecase also enables connectivity when both endpoints are behind NAT gateways. Such usecase should allow for seamless connectivity even as Endpoints roam, in being or away from gateways.

In a hub and spoke topology when two end-points communicate, they must use a mechanism for authentication, such that they do not expose them to impersonation by the other spoke endpoint.

2.2. Gateway-to-Gateway AD VPN Use Case

A typical Enterprise traffic model is Hub and Spoke, with the Gateways connecting to each other using IPsec tunnels.

However for the voice and other rich media traffic that occupies a lot of bandwidth and the traffic tromboning to the Hub can create traffic bottlenecks on the Hub and can lead to a increase cost. It is for this purpose Spoke-to-Spoke tunnels are dynamically created and torn-down.

The Spoke Gateways can themselves come up and down, getting different IP addresses in the process, making th static configuration impossible.

Also for the reasons of cost and manual error reduction, it is desired there be minimal or even no configuration on the Hub as a new Spoke Router is added or removed.

In a hub and spoke topology when two spoke gateways communicate, they must use a mechanism for authentication, such that they do not expose them to impersonation by the other gateways spoke.

2.3. Endpoint-to-Gateway AD VPN Use Case

An endpoint should be able to use the most efficient gateway as it roams in the internet.

A mobile user roaming on the Internet may connect to a gateway, which because of roaming is no longer the most efficient gateway to use (reasons could be cost/ efficiency/ latency or some other factor). The mobile user should be able to discover and then connect to the current most efficient gateway without having to reinitiate the connection.

3. Inadequacy of Existing Solutions

Several solutions exist for the problems described above. However, none of these solutions is adequate, as described here.

3.1. Exhaustive Configuration

One simple solution is to configure all gateways and endpoints in advance with all the information needed to determine which gateway or endpoint is optimal and to establish an SA with that gateway or endpoint. However, this solution does not scale in a large network with hundreds of thousands of gateways and endpoints, especially when multiple organizations are involved and things are rapidly changing (e.g. mobile endpoints). Such a solution is also limited by the smallest endpoint/ gateway, as the same exhaustive configuration is to be applied on all endpoints/ gateways. A more dynamic, secure and scalable system for establishing SAs between gateways is needed.

3.2. Star Topology

The most common way to address this problem today is to use what has been termed a "star topology". In this case one or a few gateways are defined as "Hub gateways", while the rest of the systems (whether endpoints or gateways) are defined as "spokes". The spokes never connect to other spokes. They only open tunnels with the core gateways. Also for a large number of gateways in one administrative domain, one gateway may be defined as the core, and the rest of the gateways and remote access clients connect only to that gateway.

This solution however does not work when the spokes, get dynamic IP address which the "core gateways" cannot be configured with. It is also desired that there is minimal to no configuration on the Hub as the number of spokes increases and new spokes are added and deleted randomly.

Another problem with stars and trunks is that it creates a high load on the core gateways as well as on the trunk connection. This load is both in processing power and in network bandwidth. A single packet in the trunk scenario can be encrypted and decrypted three times. It would be much preferable if these gateways and clients could initiate tunnels between them, bypassing the core gateways. Additionally, the path bandwidth to these core gateways may be lower than that of the path between the satellites. For example, two remote access users may be in the same building with high-speed wifi (for example, at an IETF meeting). Channeling their conversation through the core gateways of their respective employers seems extremely wasteful, as well as having lower bandwidth.

The challenge is to build a large scale, IPsec protected networks that can dynamically change with minimum administrative overhead.

3.3. Proprietary Approaches

Several vendors offer proprietary solutions to these problems. However, these solutions offer no interoperability between equipment from one vendor and another. This means that they are generally restricted to use within one organization, and it is harder to move off such solutions as the features are not standardized. Besides multiple organizations cannot be expected to all choose the same equipment vendor.

4. Requirements

This section is currently being updated and hence under flux.

4.1. Gateway and End Point Requirements

1. For any network topology (whether Hub-and-Spoke or Full Mesh) Gateways/ end points MUST allow for minimal configuration changes when a new Gateway or end-point is added, removed or changed. The solution should allow for such configuration on a global basis.
2. Gateways/ end-points MUST allow IPsec Tunnels to be setup without any configuration changes, even as peer addresses gets updated every time the device comes up.
3. Gateways MUST allow tunnel binding, such that applications like Routing using the tunnels can work seamlessly without any updates to the higher level application configuration i.e. OSPF configuration.
4. In a Hub-and-Spoke topology, Spoke Gateways/ en-points MUST allow for direct communication with other Spoke Gateways/ end-points, using authentication that does not expose them to other Gateway Spoke.
5. Gateways SHOULD allow for easy handoff of sessions in case end-points are roaming and cross policy boundaries.
6. Gateways SHOULD allow for easy handoff of a session to another gateway, to optimize latency, bandwidth or other factor, based on policy.
7. Gateways/ End-points MUST be able to work, behind NAT boxes.

5. Security Considerations

The solution to the problems presented in this draft may involve dynamic updates to databases defined by RFC 4301, such as the Security Policy Database (SPD) or the Peer Authorization Database (PAD).

RFC 4301 is silent about the way these databases are populated, and it is implied that these databases are static and pre-configured by a human. Allowing dynamic updates to these databases must be thought out carefully, because it allows the protocol to alter the security policy that the IPsec endpoints implement.

One obvious attack to watch out for is stealing traffic to a particular site. The IP address for `www.example.com` is `192.0.2.10`. If we add an entry to an IPsec endpoint's SPD that says that traffic to `192.0.2.10` is protected through peer Gw-Mallory, then this allows Gw-Mallory to either pretend to be `www.example.com` or to proxy and read all traffic to that site. Updates to this database requires a clear trust model.

More to be added.

6. IANA Considerations

No actions are required from IANA for this informational document.

7. Acknowledgements

Many people have contributed to the development of this problem statement and many more will probably do so before we are done with it. While we cannot thank all contributors, some have played an especially prominent role. Yoav Nir, Yaron Scheffer, Jorge Coronel Mendoza, Chris Ulliott, and John Veizades wrote the document upon which this draft was based. Geoffrey Huang, Suresh Melam, Praveen Sathyanarayan, Andreas Steffen, and Brian Weis provided essential input.

8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

Authors' Addresses

Steve Hanna
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: shanna@juniper.net

Vishwas Manral
Hewlett-Packard Co.
19111 Pruneridge Ave.
Cupertino, CA 95113
USA

Email: vishwas.manral@hp.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

Y. Nir
Check Point
July 16, 2012

A TCP transport for the Internet Key Exchange
draft-nir-ipsecme-ike-tcp-01

Abstract

This document describes using TCP for IKE messages. This facilitates the transport of large messages over paths where fragments are either dropped, or packet loss makes them unreliable.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

The Internet Key Exchange (IKE) specified in [RFC2407] and [RFC2408], and IKEv2 as specified in [RFC5996] uses UDP to transport the exchange messages. Some of those messages may be fairly large. Specifically, the 5th and 6th messages of IKEv1 Main Mode, the first and second messages of IKEv1 Aggressive Mode, and the messages of IKEv2 IKE_AUTH exchange can become quite large, as they may contain a chain of certificates, a signature payload (called "Auth" in IKEv2), CRLs, and in the case of IKEv2, some configuration information that is carried in the CFG payload.

When such UDP packets exceed the path MTU, they get fragmented. This increases the probability of packets getting dropped, but the retransmission mechanisms in IKE (as described in section 2.1 of RFC 5996) takes care of that. More recently we have seen a number of service providers dropping fragmented packets. Firewalls and NAT devices need to keep state for each packet where some but not all of the fragments have been received. This creates a burden in terms of memory, especially for high capacity devices such as Carrier-Grade NAT (CGN) or high capacity firewalls.

The BEHAVE working group has an Internet Draft describing required behavior of CGNs ([I-D.ietf-behave-lsn-requirements]). It requires CGNs to comply with [RFC4787], which in section 11 requires NAT devices to support fragments. However, some people deploying IKE have found that some ISPs have begun to drop fragments in preparation for deploying CGNs. While we all hope for a future where all devices comply with the emerging standards, or even a future where CGNs are not required, we have to make IKE work today.

The solution described in this document is to transport the IKE messages over a TCP ([RFC0793]) rather than over UDP. IKE packets (both versions) describe their own length, so they are well-suited for transport over a stream-based connection such as TCP. The Initiator opens a TCP connection to the Responder's port 500, sends the requests and receives the responses, and then closes the connection. TCP can handle arbitrary-length messages, works well with any sized data, and is well supported by all ISP infrastructure.

1.1. Non-Goals of this Specification

Firewall traversal is not a goal of this specification. If a firewall has a policy to block IKE and/or IPsec, hiding the IKE exchange in TCP is not expected to help. Some implementations hide both IKE and IPsec in a TCP connection, usually pretending to be HTTPS by using port 443. This has a significant impact on bandwidth and gateway capacity, and even this is defeated by better firewalls.

SSL VPNs tunnel IP packets over TLS, but the latest firewalls are also TLS proxies, and are able to defeat this as well.

This document is not part of that arms race. It is only meant to allow IKE to work When faced with broken infrastructure that drops large IP packets.

1.2. Conventions Used in This Document

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 [RFC2119].

2. The Protocol

2.1. Initiator

An Initiator MAY try IKE using TCP for any request. It opens a TCP connection from an arbitrary port to port 500 of the Responder. When the three-way handshake completes, the Initiator MUST send the request. If the Initiator knows that this request is the last request needed at this time, it SHOULD half-close the TCP connection, although it MAY wait until the last response is received. When all responses have been received, the Initiator MUST close the connection. If the peer has closed the connection before all requests have been transmitted or responded to, the Initiator SHOULD either open a new TCP connection or transmit them over UDP again.

It MUST accept responses sent over IKE within the same connection, but MUST also accept responses over other transports, if the request had been sent over them as well.

2.2. Responder

A Responder MAY accept TCP connections to port 500, and if it does, it MUST accept IKE requests over this connection. Responses to requests received over this connection MUST also go over this connection. If the connection has closed before the Responder had had a chance to respond, it MUST NOT respond over UDP, but MUST instead wait for a retransmission over UDP or over another TCP connection.

The responder MUST accept different requests on different transports. Specifically, the Responder MUST NOT rely on subsequent requests coming over the same transport. For example, it is entirely acceptable to have the first two requests on IKE Main Mode come over UDP port 500, while the last request comes over TCP, and the

following Quick Mode request might come over UDP port 4500 (because NAT has been detected).

A responder that receives an IKEv2 Initial request over any other transport MUST send an IKE_TCP_SUPPORTED notification (Section 2.5) in the Initial response. the responder MAY send this notification even if the Initial request was received over TCP.

If the responder has some requests of its own to send, it MUST NOT use a connection that has been opened by a peer. Instead, it MUST either use UDP or else open a new TCP connection to the original Initiator's TCP port 500.

The normal flow of things is that the Initiator opens a connection and closes its side first. The responder closes after sending the last response where the initiator has already half-closed the connection. If, however, a significant amount of time has passed, and neither new requests arrive nor the connection is closed by the initiator, the Responder MAY close or even reset the connection.

This specification makes no recommendation as to how long such a timeout should be, but a few seconds should be enough.

2.3. Transmitter

The transmitter, whether an initiator transmitting a request or a responder transmitting a response MUST NOT retransmit over the same connection. TCP takes care of that. It SHOULD send the IKE header and the IKE payloads with a single command or in rapid succession, because the receiver might block on reading from the socket.

2.4. Receiver

The IKE header is copied from RFC 5996 below for reference:

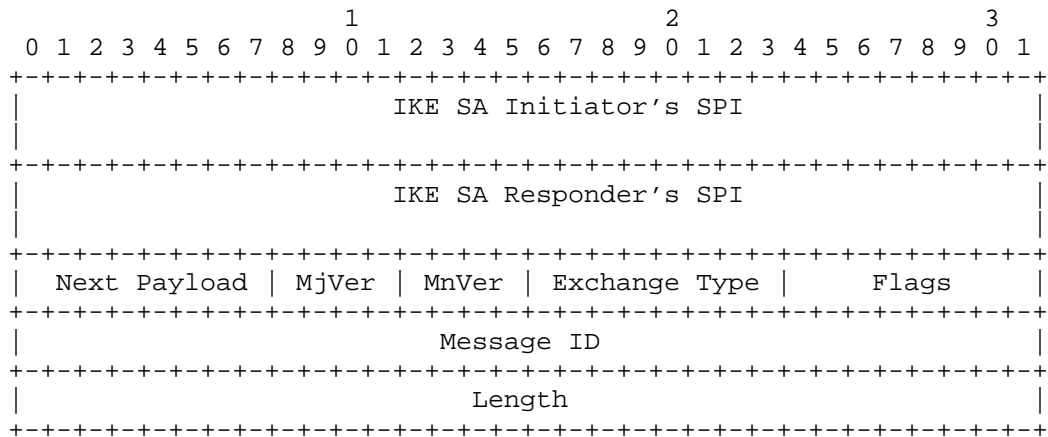


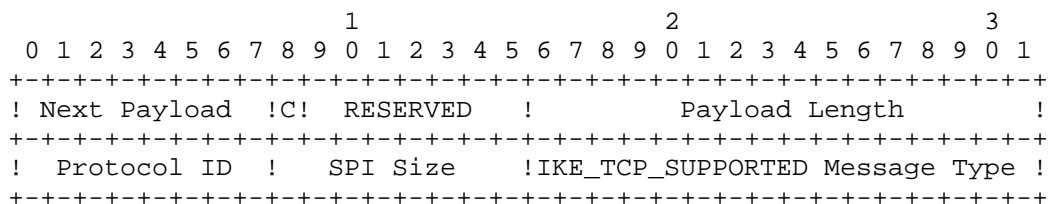
Figure 1: IKE Header Format

The receiver MUST first read in the 28 bytes that make up the IKE header. The Responder then subtracts 28 from the length field, and reads the resulting number of bytes. The combined message, comprised on 28 header bytes and whatever number of payload bytes is processed the same way as regular UDP messages. That includes retransmission detection, with one slight difference: if a retransmitted request is detected, the response is retransmitted as well, but using the current TCP connection rather than whatever other transport had been used for the original transmission of the request.

2.5. IKE_TCP_SUPPORTED Notification

This notification is sent by a responder over non-TCP transports to inform the initiator that this specification is supported.

The Notify payload is formatted as follows:



- o Protocol ID (1 octet) MUST be 0.
- o SPI Size (1 octet) MUST be zero, in conformance with section 3.10 of [RFC5996].

- o IKE_TCP_SUPPORTED Notify Message Type (2 octets) - MUST be xxxxxx, the value assigned for IKE_TCP_SUPPORTED. TBA by IANA.

3. Operational Considerations

Most IKE messages are relatively short. Quick Mode in IKEv1, and all but the IKE_AUTH exchange in IKEv2 are comprised of short messages that fit in a single packet on most networks. It is only the IKE_AUTH exchange in IKEv2, and the two last messages of Main Mode that are long. UDP has advantages in lower latency and lower resource consumption, so it makes sense to use UDP whenever TCP is not required.

The requirements in Section 2.2 mean that different requests may be sent over different transports. So the initiator can choose the transport on a per-request basis. So one obvious policy would be to do everything over UDP except the specific requests that tend to become too big. This way the first messages use UDP, and the Initiator can set up the TCP connection at the same time, eliminating the latency penalty of using TCP. This may not always be the most efficient policy, though. It means that the first messages sent over TCP are relatively large ones, and TCP slow start may cause an extra roundtrip, because the message may exceed the transmission window. An initiator using this policy MUST NOT go to TCP if the responder has not indicated support by sending the IKE_TCP_SUPPORTED notification (Section 2.5) in the Initial response.

An alternative method, that is probably easier for the Initiator to implement, is to do an entire "mission" using the same transport. So if TCP is needed and an IKE SA has not yet been created, the Initiator will open a TCP connection, and perform all 2-4 requests needed to set up a child SA over the same connection.

Yet another policy would be to begin by using UDP, and at the same time set up the TCP connection. If at any point the TCP handshake completes, the next requests go over that connection. This method can be used to auto-discover support of TCP on the responder. This is easier for the user than configuring which peers support TCP, but has the potential of wasting resources, as TCP connections may finish the three-way handshake just when IKE over UDP has finished. The requirements from the responder ensure that all these policies will work.

3.1. Liveness Check

The TCP connections described in this document are short-lived. We do not expect them to stay for the lifetime of the SA, but to get

torn down by either side within seconds of the SA being set up. Because of this, they are not well-suited for the transport of short requests such as those for liveness check.

Although liveness checks MAY be sent over TCP, this is not recommended.

4. Security Considerations

Most of the security considerations for IKE over TCP are the same as those for UDP as in RFC 5996.

For the Responder, listening to TCP port 500 involves all the risks of maintaining any TCP server. Precautions against DoS attacks, such as SYN cookies are RECOMMENDED.

5. IANA Considerations

IANA is requested to assign a notify message type from the status types range (16418-40959) of the "IKEv2 Notify Message Types" registry with name "IKE_TCP_SUPPORTED"

No IANA action is required for the TCP port, as TCP port 500 is already allocated to "ISAKMP".

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.
- [RFC2408] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

6.2. Informative References

- [I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A.,
and H. Ashida, "Common requirements for Carrier Grade NATs
(CGNs)", draft-ietf-behave-lsn-requirements-08 (work in
progress), July 2012.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7,
RFC 793, September 1981.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation
(NAT) Behavioral Requirements for Unicast UDP", BCP 127,
RFC 4787, January 2007.

Author's Address

Yoav Nir
Check Point Software Technologies Ltd.
5 Hasolelim st.
Tel Aviv 67897
Israel

Email: ynir@checkpoint.com

