

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 8, 2014

Y. Sheffer  
Porticor  
Y. Nir  
Check Point  
February 4, 2014

The AutoVPN Architecture  
draft-sheffer-autovpn-00

## Abstract

This document describes the AutoVPN architecture. AutoVPN allows IPsec security associations to be set up with no prior configuration, using the "leap of faith" paradigm. The document defines a lightweight protocol for negotiating such opportunistic encryption either directly between hosts or between two security gateways on the path.

## 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 August 8, 2014.

## Copyright Notice

Copyright (c) 2014 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

Internet-Draft

AutoVPN

February 2014

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">Terminology . . . . .</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Architecture and Protocol Overview . . . . .</a>	<a href="#">4</a>
<a href="#">4.</a>	<a href="#">Protocol Exchanges . . . . .</a>	<a href="#">6</a>
<a href="#">5.</a>	<a href="#">Message Format . . . . .</a>	<a href="#">7</a>
<a href="#">5.1.</a>	<a href="#">ICMP Encoding . . . . .</a>	<a href="#">7</a>
<a href="#">5.2.</a>	<a href="#">UDP Encoding . . . . .</a>	<a href="#">7</a>
<a href="#">5.3.</a>	<a href="#">Protocol Payloads . . . . .</a>	<a href="#">8</a>
<a href="#">5.4.</a>	<a href="#">Version Payload . . . . .</a>	<a href="#">9</a>
<a href="#">5.5.</a>	<a href="#">Nonce Payloads . . . . .</a>	<a href="#">10</a>
<a href="#">5.6.</a>	<a href="#">NAT-Detect Payload . . . . .</a>	<a href="#">10</a>
<a href="#">6.</a>	<a href="#">Error Handling and Reliability . . . . .</a>	<a href="#">10</a>
<a href="#">7.</a>	<a href="#">NAT Considerations . . . . .</a>	<a href="#">11</a>
<a href="#">8.</a>	<a href="#">IKE Protocol Considerations . . . . .</a>	<a href="#">11</a>
<a href="#">8.1.</a>	<a href="#">New IKE Payloads . . . . .</a>	<a href="#">12</a>
<a href="#">8.1.1.</a>	<a href="#">AutoVPN Nonce . . . . .</a>	<a href="#">12</a>
<a href="#">8.1.2.</a>	<a href="#">Contact Details . . . . .</a>	<a href="#">12</a>
<a href="#">8.2.</a>	<a href="#">AUTOVPN_SHARED_SECRET Notification . . . . .</a>	<a href="#">12</a>
<a href="#">9.</a>	<a href="#">Security Policy . . . . .</a>	<a href="#">12</a>
<a href="#">9.1.</a>	<a href="#">Certificate States . . . . .</a>	<a href="#">12</a>
<a href="#">9.2.</a>	<a href="#">Certificate Rollover and Permanent Association . . . . .</a>	<a href="#">14</a>
<a href="#">9.3.</a>	<a href="#">Certificate Conflicts . . . . .</a>	<a href="#">14</a>
<a href="#">9.4.</a>	<a href="#">Fallback to Clear . . . . .</a>	<a href="#">15</a>
<a href="#">10.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">15</a>
<a href="#">11.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">15</a>
<a href="#">12.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">15</a>
<a href="#">13.</a>	<a href="#">References . . . . .</a>	<a href="#">15</a>
<a href="#">13.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">15</a>
<a href="#">13.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">16</a>
<a href="#">Appendix A.</a>	<a href="#">Change Log . . . . .</a>	<a href="#">16</a>
<a href="#">A.1.</a>	<a href="#">-00 . . . . .</a>	<a href="#">16</a>
<a href="#">Appendix B.</a>	<a href="#">Implementation Considerations . . . . .</a>	<a href="#">17</a>
<a href="#">B.1.</a>	<a href="#">Address Authorization . . . . .</a>	<a href="#">17</a>
<a href="#">B.2.</a>	<a href="#">Multiple Interfaces and Alternative Gateways . . . . .</a>	<a href="#">17</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">17</a>

## [1.](#) Introduction

In the last few years, there have been several attempts to define an opportunistic encryption architecture, where network traffic is confidentiality-protected, even in the absence of proper authentication of the peers. This protection is often accompanied by continuity of identity, i.e. although the identity may not be

authenticated, it is verified to remain unchanged between multiple protocol sessions, and over long periods (days/weeks). This initial protection may be enhanced at a later stage, as peers gain stronger trust in each other's identity. A term which is often used to denote this policy is "leap of faith".

In the IPsec space, these attempts culminated in the BTNS (Better Than Nothing Security) working group's specifications. The BTNS working group produced a number of documents, including [[RFC5386](#)] and [[RFC5387](#)]. In addition, the earlier [[RFC4322](#)] describes Opportunistic Encryption, as implemented in various dialects of Linux (the history of Linux OE is summarized in a long post by Paul Wouters [[oe-history](#)]). However these specifications focus on the architectural IPsec implications, and provide insufficient context to implement the behavior described in the current document. "Leap of faith" has never been fully specified in the IPsec context, or when specified, assumes mechanisms that are still not widely deployed.

Similarly to many security architectures, a well designed opportunistic encryption solution requires both a robust protocol, and a user interaction component that allows the user to understand the exact security guarantees available at any time, so that the user may add external inputs about trustworthiness of communication peers while staying away from the "just press OK" mentality.

This document describes the AutoVPN architecture, an opportunistic encryption extension to the Internet Key Exchange v2 (IKEv2 - [[RFC5996](#)]) for IPsec VPN.

Some of the requirements behind this protocol are:

- o It should be suitable for business-to-business traffic, and therefore for deployment on the open Internet.
- o It should be robust, efficient and network friendly enough to be enabled by default.

- o It should be deployable on (existing) security gateways, rather than requiring changes to hosts.
- o It should also work on hosts that are not protected by gateways, i.e. hosts that are themselves IPsec endpoints.
- o It should require zero configuration. Some limited level of security should be provided by devices which are not configured.
- o After-the-fact security: the security guarantees can be improved at a later time, possibly using out-of-band means.

- o The protocol should coexist with regular IPsec, with no degradation in security.
- o The protocol should provide the best possible security given the imperfections of today's Internet. In particular, it should not rely on the deployment of DNS Security, anti spoofing mechanisms or routing security.
- o Small gateways, as well as software VPN clients, are often behind NAT. This scenario should be supported.

## 2. Terminology

We use the term "initiator" for the gateway through which came the original traffic. The gateway may not be the initiator of the new protocol described below. The other gateway is the "responder". Note that these terms correspond to the gateways' behavior with respect to IKE negotiation.

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](#)].

## 3. Architecture and Protocol Overview

The protocol creates an IPsec tunnel to protect traffic which would otherwise be transmitted in the clear. What follows is a high level description of the sequence of operations.

We use H1 and H2 to denote two hosts (endpoints), and G1 and G2 to denote two IPsec gateways, protecting H1 and H2 respectively. This setup is shown in Figure 1 below. The solution described here is also applicable when one or both hosts is collocated with its respective gateway. Unfortunately it cannot be optimized for these cases, since source IP addresses can always be spoofed.

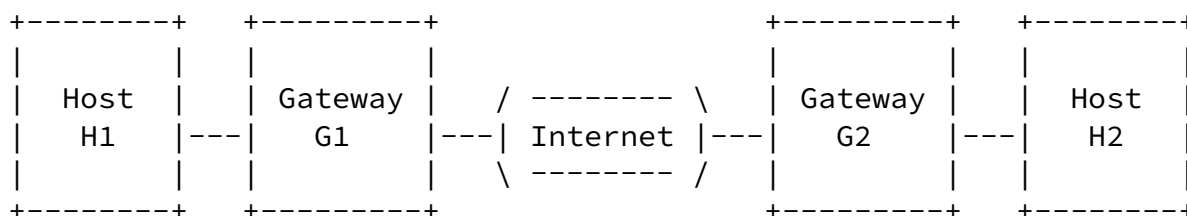


Figure 1: Deployment Architecture

Initially, only H1 knows of H2's address. The protocol below allows both intervening gateways to discover each other, and to gain

assurance that each one is on-path of the opposite host, i.e. it can see traffic addressed to the respective host, and can respond to such traffic.

We assume that both G1 and G2 contain access control functionality, as required by the IPsec architecture, and that both allow some clear traffic between H1 and H2.

The message sequence below is motivated to a great extent by the need to cater to NAT devices in front of G1, the original initiator. It is assumed that correctly implemented NAT devices will perform correct reverse translation of ICMP messages. However we cannot assume that they handle correctly ICMP messages of an unknown type.

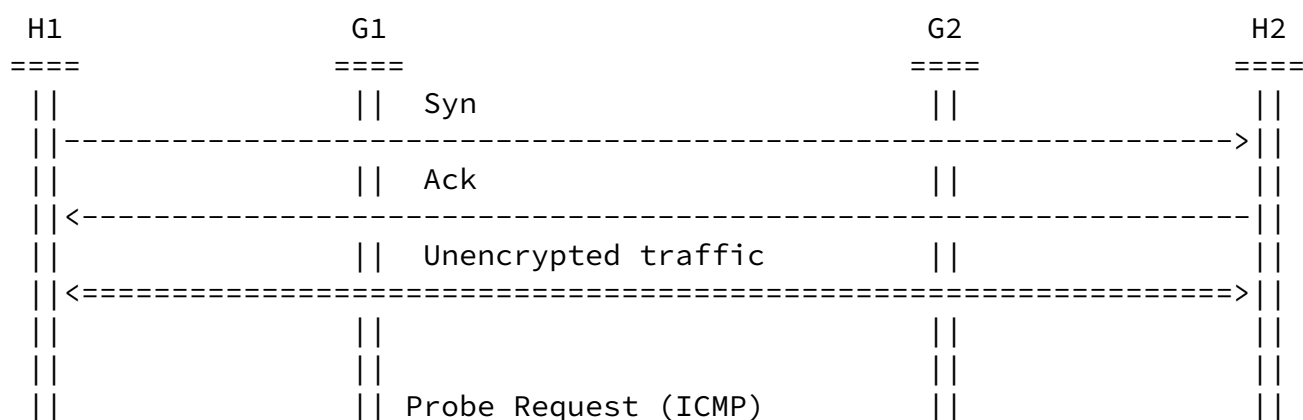
Another major consideration is which side should drive the exchange. We have chosen the responder side (G2), since in an Internet where most traffic uses HTTP, the responder side knows best which traffic should be protected.

Lastly, we could have saved one round trip by allowing G2 to spoof H2's address. We believe this would have been ill advised.

The flow of messages is depicted in Figure 2.

- o H1 creates a network connection to H2, for example by sending a TCP SYN packet. H2 replies normally to H1.
- o G2 intercepts the reply packet, but lets it pass through. The connection proceeds normally, possibly including data packets.
- o G2 sends a Probe Request message, addressed to H1.
- o G1 intercepts the Probe Request, does not forward it, and sends a Probe Response, addressed to H2. Note that if H1 is NOT protected by a gateway, it will receive the Probe Request message and therefore the message should be designed to have no effect on innocent receivers.
- o G2 intercepts the Probe Response, does not forward it, and sends a Probe Complete, addressed to G1.
- o G1 now initiates an IKE\_SA\_INIT exchange to G2. This message includes payloads that can be correlated with the previous messages.
- o G1 and G2 negotiate an IPsec SA, potentially for all traffic between H1 and H2.

- o Both G1 and G2 now move the traffic between H1 and H2 into the new SA.



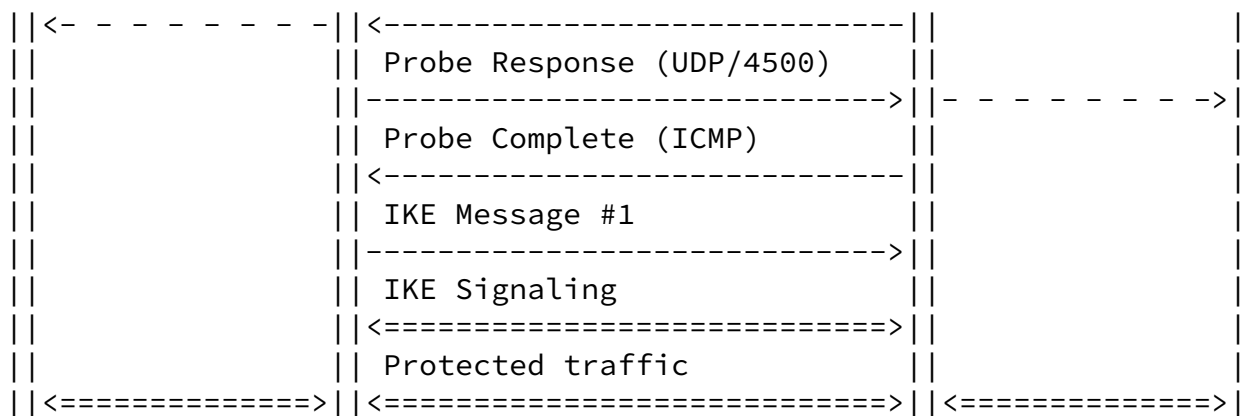


Figure 2: Message Sequence

#### 4. Protocol Exchanges

AutoVPN consists of a 3-way probing protocol, followed by a slightly extended IKEv2 exchange.

The normal execution sequence of the protocol is as follows. G2 generates a fresh, randomly generated value Nonce-R, and sends to H1:

Probe Request: Version, Nonce-R, NAT-Detect

G1 intercepts the received message and does not forward it to H1. G1 MAY verify that the message corresponds to an ongoing connection, using the packet fragment contained in the ICMP envelope. G1 generates a fresh, random Nonce-I and sends to H2:

Probe Response: Version, Nonce-I, Nonce-R

where the content of Nonce-R is copied from the request. G2 intercepts this message, and does not forward it to H2. G2 MUST

verify that Nonce-R is valid, and silently ignore the message otherwise. G2 replies with:

Probe Complete: Version, Nonce-I, Nonce-R

G1 MUST check the validity of Nonce-I and Nonce-R. Finally, G1 sends an IKEv2 IKE\_SA\_INIT message to G2, containing a copy of the received Nonce-R.

## [5.](#) Message Format

The AutoVPN protocol messages consist of a sequence of type-length-value (TLV) payloads. The messages are encoded in two different base protocols: ICMP and UDP over port 4500.

The Probe Request and Probe Complete messages MUST be encoded within ICMP. The Probe Response message MUST be encoded within UDP.

### [5.1.](#) ICMP Encoding

Each payload is encoded as an ICMP Extension Object, as per [\[RFC4884\]](#). ICMP Error messages contain a copy of (part of) the original packet, and this is used to associate the ICMP message with the original clear traffic. ICMP header fields are populated as follows:

- o Type is "Parameter Problem".
- o Code is the value 1.
- o The Checksum field MUST be computed, as per [\[RFC0792\]](#).
- o The new Length field is defined in [\[RFC4884\]](#).

In the Extension Object Headers, Class-Num is TBD by IANA, and C-Type is the Type value defined for each payload below.

### [5.2.](#) UDP Encoding

In this encoding, all payloads are simply concatenated following the Preamble. Each payload is preceded by a payload header, as defined in [Section 5.3](#).

The generic Preamble format is described in the next figure.



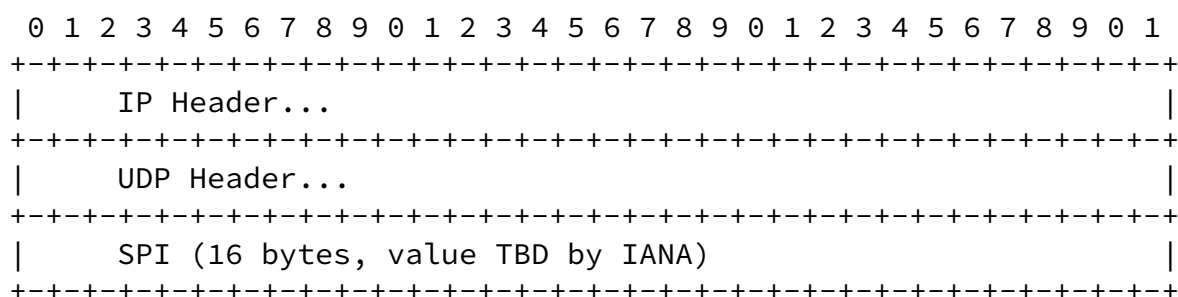


Figure 3: AutoVPN Preamble

The Probe Response protocol message has 4500 as its destination port. The protocol reuses the IKE/IPsec port 4500, however it is neither IKE nor IPsec. All three can coexist, and are distinguished using the SPI value. The specific SPI value will be allocated out of the "reserved" SPI space.

### 5.3. Protocol Payloads

An AutoVPN payload is encoded as an ICMP Extension Object or within a UDP message. When using UDP, the generic payload header is described in the next figure:

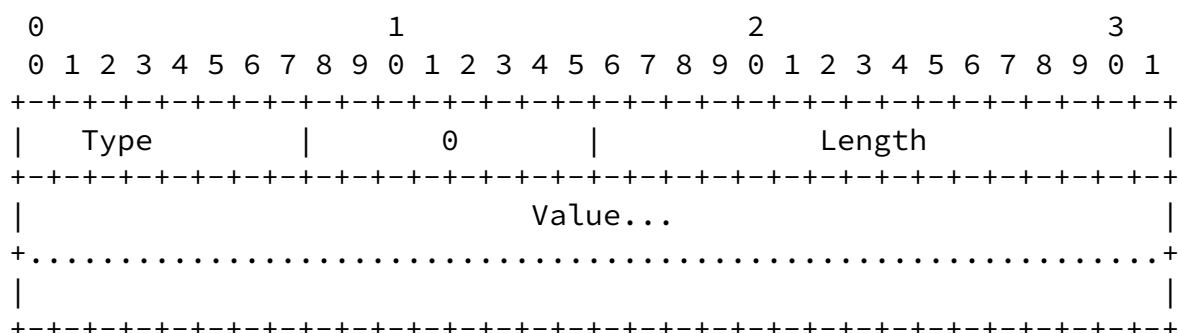


Figure 4: Payload Header

Type:

One of the payload types listed below.

Length:

The payload length in octets, including this header.

The following payload types are defined:

Name	Value	Definition
Unused	0	
Version	1	Generic information about the current message
Nonce-I	2	Initiator's nonce
Nonce-R	3	Responder's nonce
NAT-Detect	4	NAT detection information
	4-127	Reserved to IANA
	128-255	Reserved for private use

#### 5.4. Version Payload

This payload is formatted as follows:

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			
Version	Message Type	Reserved	Vendor ID
~			
+++++			
~			
+++++			

Figure 5: Version Payload

The header contains the following fields:

Version:

MUST be 0x01 for this version of the protocol.

Message Type:

1 for Probe Request, 2 for Probe Response, 3 for Probe Complete.  
Other values are reserved to IANA.

Reserved:

MUST be sent as 0, and ignored by the receiver.

Internet-Draft

AutoVPN

February 2014

#### Vendor ID:

This field is optional and of variable length, possibly 0. It MAY contain a string uniquely identifying the vendor (e.g. "example.com"), or a binary string that is statistically unique (e.g. the SHA-1 hash of "we support the XX extension").

### [5.5.](#) Nonce Payloads

Nonces are random or unpredictable values that enable the entity that generated them to recognize them as valid when it receives them again. Nonces MAY be used to encode state, in order to enable stateless implementations of this protocol. The length of each nonce (excluding the payload header) MUST be between 8 and 64 octets, inclusive.

One possible way to construct the nonce is

```
key-ID || HMAC-SHA256(K, gateway-IP || packet-fragment)
```

where K is a secret key known only to the sender, and key-ID identifies the key, enabling smooth roll-over of keys. Packet-fragment is the same portion of the packet as returned in an ICMP response, i.e. the IP header and the 8 octets that follow it.

### [5.6.](#) NAT-Detect Payload

This payload consists of a 4-octet obfuscated IPv4 address, followed by a 2-octet port number. The address is obfuscated by a XOR operation with 0x0F0F0F0F, with the intention of defeating over-eager NAT devices which might try to rewrite the packet. The address and port are the source address/port of the original (clear) packet, as seen by the remote gateway (G2). The IP protocol (e.g. UDP or TCP) is inferred from the packet fragment included in the ICMP message containing this payload.

## [6.](#) Error Handling and Reliability

The AutoVPN protocol is UDP and ICMP based, and therefore per-message reliability is not guaranteed. Both sides MAY retransmit the ICMP and UDP messages, but MUST NOT do so more than twice (total of 3 messages). The gateway (G2) that sends the first ICMP message MUST NOT retry a particular peer more than once every 24 hours.

The protocol does not include any error messages. If a peer does not accept a particular message for any reason, it MUST silently drop it. For forward compatibility, a receiver SHOULD process incoming

messages even if they contain payloads that it does not understand, and SHOULD ignore these payloads.

## 7. NAT Considerations

The current version of the protocol allows both sides to detect a NAT being performed between the gateways. Detection takes place during the probing phase. However this scenario raises several issues, which require further investigation before a useful solution can be proposed:

- o If traffic from a host which is behind NAT (H1) is inserted directly into an IPsec tunnel, it will emerge as-is on the other side, and the receiving host might see a non-routable [[RFC1918](#)] source address.
- o The initiating gateway may not have enough information to formulate its IKE Traffic Selector payload (TSi).

A solution that can be considered is for G1 to request a Tunnel Inner Address using an IKE Configuration Payload, and to perform NAT on traffic originating from H1 so that it appears to be sourced from that address. One of the issues with this solution is that the initial clear connection will necessarily be broken because of the address change.

We note that the protocol can be implemented correctly on a gateway that performs the NAT function itself.

## 8. IKE Protocol Considerations

The AutoVPN protocol imposes a few requirements on the IKE peers:

- o Both peers MUST use a certificate to authenticate. In many cases this is expected to be a self-signed certificate. But see also [Section 9.2](#).
- o The peers MUST NOT negotiate any IPsec protocol, other than ESP in tunnel mode.
- o Each peer MUST offer only a single IP address in its negotiated traffic selector. This IP address MUST be identical to the one the gateway has proven authorization for. This MUST also be validated by the opposite peer. Per policy, traffic selectors MAY be even narrower, e.g. referring to specific protocol ports.

## [8.1.](#) New IKE Payloads

This protocol defines several new IKE payloads.

### [8.1.1.](#) AutoVPN Nonce

This payload has the payload type TBD by IANA. It MUST only be used in the first message of the IKE\_SA\_INIT exchange. The payload contains an exact copy of the Nonce-R AutoVPN payload, without the AutoVPN payload header.

### [8.1.2.](#) Contact Details

This payload has the payload type TBD by IANA. It SHOULD be sent by both peers during the IKE\_AUTH exchange. The payload contains a human readable UTF-8 string which is designed to assist the person managing the opposite protocol peer in verifying the sender's true identity. An example string is:

This gateway is operated by Example Inc. To validate our identity, you may wish to obtain our public key's fingerprint from our Web site, at <https://www.example.com/autovpn>. Or you may wish to contact the network administrator at 1-616-555-1212 to get the fingerprint. Please compare this value with the fingerprint value displayed by your gateway.

For obvious security reasons, this string MUST be rendered as plain text, and in particular MUST NOT be rendered as HTML.

## [8.2.](#) AUTOVPN\_SHARED\_SECRET Notification

This notification, whose value is TBD by IANA, contains no data. It signifies that an AutoVPN shared secret MUST be created by the two IKE peers. See [Section 9.2](#) for details.

## [9.](#) Security Policy

This section describes the AutoVPN security policy, and should be viewed as an extension of [[RFC4301](#)].

### [9.1.](#) Certificate States

AutoVPN defines a state for each peer gateway's certificate. A certificate may be in one of the following states (Figure 6):

- o Unknown. This may also be a certificate which had been manually removed, through a manual operation or for a number of reasons listed below.

- o Known but unverified. A DN is associated with a certificate's fingerprint, and additional information may be available ("contact details"). When not used, such certificates may be deleted from the table, after some site-specific timeout. It is RECOMMENDED that this timeout be larger than 7 days.
- o Trusted identity. The administrator can manually mark a certificate as Trusted. Alternatively, the certificate may have been signed by a trusted third party.
- o Untrusted identity. The administrator can manually mark a certificate as Untrusted, if he or she manually checks the certificate's fingerprint and detects a mismatch against an advertised value.
- o Managed. These certificates belong to peers that are part of the same managed VPN. They are not further discussed in this document.

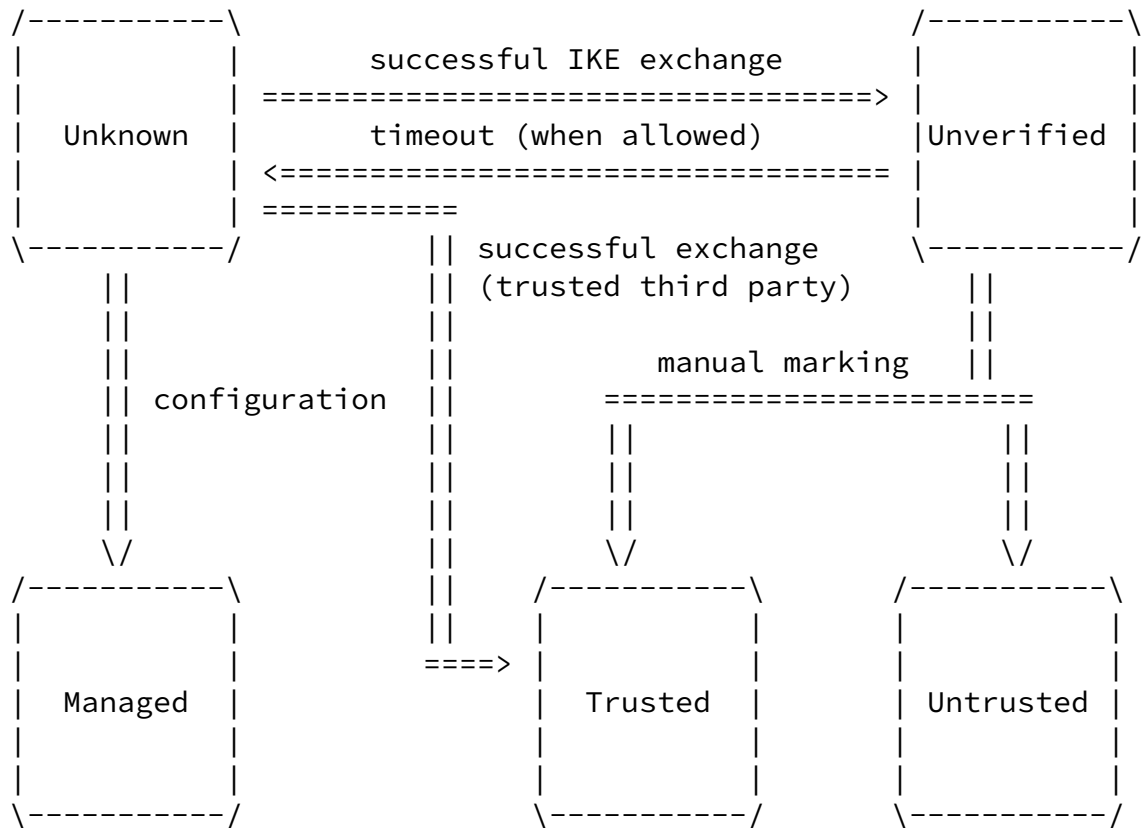


Figure 6: Certificate States

If the gateway detects that a peer's certificate has been explicitly revoked, it **MUST** delete this certificate from the table.

A PAD entry may exist for certificates in the Unverified, Managed or Trusted states. A PAD entry **MUST NOT** exist for a certificate in the Untrusted state, and IKE exchanges with peers presenting such certificates **MUST** be rejected, regardless of who initiated the exchange. When a certificate is deleted (whether manually or automatically) or marked as Untrusted, the associated PAD entry **MUST** be deleted.

When locating the peer, only the DN should be used. The peer's IP address **MUST NOT** be used, to allow peers to change their address.

## [9.2.](#) Certificate Rollover and Permanent Association

In the absence of a certificate rollover mechanism, it would be impossible to distinguish between a legitimate peer presenting a new certificate and a MITM attacker. Therefore, AutoVPN gateways **MUST** support the shared secret mechanism described here.

As noted above, the information about a peer's certificate will normally time-out and be deleted. However any of the gateways can choose a convenient time to "promote" the association between the gateways, by triggering the creation of a shared secret. This secret never expires, other than through manual deletion on both peers.

The shared secret is associated with the pair of gateway identities, specifically with the IDi, IDr payloads exchanged between the gateways. Once a shared secret is established, both gateways **MUST** use it with the associated peer, in preference to certificate-based or other forms of authentication. A shared secret **MAY** be initiated for a peer in Unverified or Trusted state. On each gateway, the shared secret is associated with the peer gateway's certificate, and both **MUST NOT** be timed out regardless of the certificate's trust state.

The initiating gateway, which may be an IKE initiator or responder, **MAY** send the AUTOVPN\_SHARED\_SECRET notification at any time. The shared secret is the value

prf+(SK\_d, "shared secret for AutoVPN")

where SK\_d is the derivation key of the current IKE SA. The literal string is represented in ASCII, with no zero terminator.

## [9.3.](#) Certificate Conflicts

A certificate conflict may be detected during the IKE exchange. This happens when an AutoVPN peer presents a certificate whose DN matches the DN of a known AutoVPN certificate, but which is different from

that certificate. In such cases the new peer **MUST** be rejected, with the notification AUTHENTICATION\_FAILED.

As a result, barring incorrect configuration, the certificate table



can never contain multiple certificates with the same DN.

#### [9.4.](#) Fallback to Clear

In some cases it may be desirable to allow fallback to clear traffic in cases where an IKE/IPsec association cannot be established, even when the peer is known. This is left to local policy, and SHOULD be configurable on the gateway. Such configuration MAY take the trust level of the peer gateway into account.

Moreover, some policies may prefer to send traffic unprotected, even when an IKE SA can be established, and then renegotiate an IPsec SA following some manual action. One possible way of doing this is using the mechanism described in [[RFC6023](#)].

#### [10.](#) IANA Considerations

TBD.

#### [11.](#) Security Considerations

TBD.

#### [12.](#) Acknowledgements

A proof of concept implementation of this protocol was created by Michael Rogovin at Check Point, and we would like to acknowledge his contribution.

#### [13.](#) References

##### [13.1.](#) Normative References

- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, [RFC 792](#), September 1981.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", [BCP 5](#), [RFC 1918](#), February 1996.
- [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.
- [RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", [RFC 4884](#), April 2007.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", [RFC 5996](#), September 2010.

### [13.2](#). Informative References

- [RFC4322] Richardson, M. and D. Redelmeier, "Opportunistic Encryption using the Internet Key Exchange (IKE)", [RFC 4322](#), December 2005.
- [RFC5386] Williams, N. and M. Richardson, "Better-Than-Nothing Security: An Unauthenticated Mode of IPsec", [RFC 5386](#), November 2008.
- [RFC5387] Touch, J., Black, D., and Y. Wang, "Problem and Applicability Statement for Better-Than-Nothing Security (BTNS)", [RFC 5387](#), November 2008.
- [RFC6023] Nir, Y., Tschofenig, H., Deng, H., and R. Singh, "A Childless Initiation of the Internet Key Exchange Version 2 (IKEv2) Security Association (SA)", [RFC 6023](#), October 2010.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", [RFC 6480](#), February 2012.
- [oe-history] Wouters, P., "History and implementation status of Opportunistic Encryption for IPsec", September 2013, <<http://nohats.ca/wordpress/blog/2013/09/12/history-and-implementation-status-of-opportunistic-encryption-for-ipsec/>>.

## [Appendix A](#). Change Log

### [A.1](#). -00

Initial version.

Internet-Draft

AutoVPN

February 2014

## [Appendix B](#). Implementation Considerations

### [B.1](#). Address Authorization

Address authorization SHOULD be maintained separately from peer identity. Timing out authorization data (as proposed in [\[RFC4322\]](#)) is risky, since the authorization protocol allows a MITM, and also exposes clear traffic. This issue is TBD, and for now, authorization will only be removed when a peer is deleted.

We should look at alternative ways to prove address ownership. For example, if the gateway G1 can prove its ownership of a certain address range, it might send an RPKI [\[RFC6480\]](#) certificate to that effect, plus proof of possession in IKE\_AUTH. The peer gateway G2 might then decide to allow a wider traffic selector including all of G1's addresses, instead of just H1.

Also TBD are the IPsec policy implications, within the framework of [\[RFC4301\]](#), Sec. 4.4.3.

### [B.2](#). Multiple Interfaces and Alternative Gateways

We might want to support multiple gateway addresses in the probing protocol, so we can have high quality connectivity without resorting to "fallback to the clear" (i.e. have very long timeouts, measured in days). On the other hand, maybe MOBIKE does the work in IKEv2.

#### Authors' Addresses

Yaron Sheffer  
Porticor

Email: [aronf.ietf@gmail.com](mailto:aronf.ietf@gmail.com)

Yoav Nir  
Check Point Software Technologies Ltd.  
5 Hasolelim st.  
Tel Aviv 6789735  
Israel

Email: [ynir@checkpoint.com](mailto:ynir@checkpoint.com)

Sheffer & Nir

Expires August 8, 2014

[Page 17]