Global Journal of Computer Science and Technology
Network, Web & Security
Volume 13 Issue 9 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Design and Implementation of Mobility for Virtual Private Network Users

By Md. Hashmathur Rehman

Abstract - Virtual Private Network framework provides Confidentiality, Integrity, Availability, Authentication and Anti- Replay services to the packets travelling through the shared medium like Internet. With the latest Advancement in the technology, Internet is available to the users thru all means like Wireless networks, GPRS, Satellite. When the VPN user roams or switches from one network to other, the IP address gets changed and VPN connection is tear down. The user has to again initiate the VPN connection whenever the network is switched. This paper present outcome of research project aimed at solving the mobility problems faced by roaming VPN users.

Keywords : VPN, IPsec, AH, ESP, authentication algorithms, encryption algorithms.

GJCST-E Classification : C.2.1

Design and Implementation of Mobility for Virtual Private Network Users

Md. Hashmathur Rehman

Abstract - Virtual Private Network framework provides Confidentiality, Integrity, Availability, Authentication and Anti-Replay services to the packets travelling through the shared medium like Internet. With the latest Advancement in the technology, Internet is available to the users thru all means like Wireless networks, GPRS, Satellite. When the VPN user roams or switches from one network to other, the IP address gets changed and VPN connection is tear down. The user has to again initiate the VPN connection whenever the network is switched. This paper present outcome of research project aimed at solving the mobility problems faced by roaming VPN users.

Keywords : VPN, IPsec, AH, ESP, authentication algorithms, encryption algorithms.

I. Introduction

IP packets doesn't have any security when they travel through shared medium like internet. It needs security services like Confidentiality, Integrity, Authentication [6, 7]. Confidentiality is provided by encryption algorithms like DES, 3DES, AES. Integrity is provided by Hash Algorithms like MD5, SHA-1, SHA-2. Authentication is provided by preshared key mechanism or by using public key infrastructure like RSA (Rivest Shamir-Adleman) [8, 12]. These services are provided by maintaining shared state between the source and the destination of an IP datagram.

The protocol to establish this state dynamically is "Internet Key Exchange (IKE)" [1, 2]. IKE performs mutual authentication between two parties and establishes an IKE security association (SA) that includes shared secret information that can be used to efficiently establish SAs for Encapsulating Security Payload [ESP] or Authentication Header [AH] and a set of cryptographic algorithms to be used by the SAs to protect the traffic that they carry[1, 2].

II. VPNs, IKEv2

All IKE communications consist of pairs of messages: a request and are sponse. The pair is called an "exchange" and is sometimes called a "request/-response pair" [1]. The first exchanges of messages establishing an IKE SA are called the IKE_SA_INIT and IKE_AUTH exchanges; subsequent IKE exchanges are called the CREATE_CHILD_SA or INFORMATIONAL exchanges [2]. In the common case, there is a single IKE_SA_INIT exchange and a single IKE_AUTH exchange (a total of four messages) to establish the IKE SA and the first Child SA.

The first exchange of an IKE session, IKE_SA_INIT, negotiates security parameters for the IKE SA, sends nonces and sends Diffie-Hellman values [2].

The second exchange, IKE_AUTH, transmits identities, proves knowledge of the secrets corresponding to the two identities and sets up an SA for the first (and often only) AH or ESP Child SA (unless there is failure setting up the AH or ESP Child SA, in which case the IKE SA is still established without the Child SA) [2].

The types of subsequent exchanges are CREATE_CHILD_SA (which creates a Child SA) and INFORMATIONAL (which deletes an SA, reports error conditions, or does other housekeeping) [2]. Every request requires are sponse. An INFORMATIONAL request with no payloads (other than the empty Encrypted payload required by the syntax) is commonly used as a check for liveness. These subsequent exchanges cannot be used until the initial exchanges have completed [2].

a) Usage Scenarios

i. *Security Gateway to Security Gateway in Tunnel Mode*

IKE is used to negotiate ESP or AH SAs in a number of different scenarios, each with its own special requirements.

```
                +-+-+-+-+-+              +-+-+-+-+-+
                |          | IPsec       |          |
Protected       |Tunnel    | tunnel      |Tunnel    |     Protected
Subnet     <-->|Endpoint  |<---------->|Endpoint  |<--> Subnet
                |          |             |          |
                +-+-+-+-+-+              +-+-+-+-+-+
```

Figure 1 : Security Gateway to Security Gateway Tunnel

In this scenario, neither endpoint of the IP connection implements IPsec, but network nodes between them protect traffic for part of the way. Protection is transparent to the end points and depends on ordinary routing to send packets through the tunnel endpoints for processing. Each endpoint would announce the set of addresses "behind" it, and packets would be sent in tunnel mode where the inner IP header

Author : MobileIron Softwares India Pvt.
E-mail : ltdhashmath@gmail.com

would contain the IP addresses of the actual end points [2].

ii. *Endpoint-to-Endpoint Transport Mode*

```
+-+-+-+-+                                        +-+-+-+-+-+
|       |            IPsec transport             |         |
|Protected|          or tunnel mode SA           |Protected|
|Endpoint |<------------------------------------->|Endpoint |
|       |                                        |         |
+-+-+-+-+                                        +-+-+-+-+-+
```

Figure 2 : Endpoint to Endpoint

In this scenario, both endpoints of the IP connection implement IPsec, as required of hosts in [IPSECARCH]. Transport mode will commonly be used with no inner IP header. A single pair of addresses will be negotiated for packets to be protected by this SA. The seend points MAY implement application-layer access controls based on the IPsec authenticated identities of the participants [2].

iii. *Endpoint to Security Gateway in Tunnel Mode*

```
+-+-+-+-+-+                        +-+-+-+-+-+
|         |       IPsec            |         |     Protected
|Protected|       tunnel           |Tunnel   |     Subnet
|Endpoint |<---------------------->|Endpoint |<--- and/or
|         |                        |         |     Internet
+-+-+-+-+-+                        +-+-+-+-+-+
```

Figure 3 : Endpoint to Security Gateway Tunnel

In this scenario, a protected endpoint (typically a portable roaming computer) connects back to its corporate network through an IPsec-protected tunnel. The packets will use tunnel mode. On each packet from the protected endpoint, the outer IP header will contain the source IP address associated with its current location (i.e., the address that will get traffic routed to the endpoint directly), while the inner IP header will contain the source IP address assigned by the security gateway (i.e., the address that will get traffic routed to the security gateway for forwarding to the endpoint). The outer destination address will always be that of the security gateway while the inner destination address will be the ultimate destination for the packet [2].

b) *Cryptographic Algorithm Negotiation*

The payload type known as "SA" indicates a proposal for a set of choices of IPsec protocols (IKE, ESP, or AH) for the SA as well as cryptographic algorithms associated with each protocol [8, 9].

An SA payload consists of one or more proposals. Each proposal includes one protocol. Each protocol contains one or more transforms -- each specifying a cryptographic algorithm. Each transform contains zero or more attributes (attributes are needed only if the Transform ID does not completely specify the cryptographic algorithm) [2, 8].

This hierarchical structure was designed to efficiently encode proposals for cryptographic suites when the number of supported suites is large because multiple values are acceptable for multiple transforms. The responder MUST choose a single suite, which may be any subset of the SA proposal following the rules below [9].

Each proposal contains one protocol. If a proposal is accepted, the SA response MUST contain the same protocol. The responder MUST accept a single proposal or reject them all and return an error. The error is given in a notification of type NO_PROPOSAL_CHOSEN [2, 9].

Each IPsec protocol proposal contains one or more transforms. Each transform contains a Transform Type. The accepted cryptographic suite MUST contain exactly one transform of each type included in the proposal [2]. For example: if an ESP proposal includes transforms ENCR_3DES, ENCR_AES w/key size 128, ENCR_AES w/key size 256, AUTH_HMAC_MD5, and AUTH_HMAC_SHA, the accepted suite MUST contain one of the ENCR_ transforms and one of the AUTH_ transforms. Thus, six combinations are acceptable [2, 8].

If an initiator proposes both normal ciphers with integrity protection as well as combined-mode ciphers, then two proposals are needed. One of the proposals includes the normal ciphers with the integrity algorithms for them and the other proposal includes all the combined-mode ciphers without the integrity algorithms (because combined-mode ciphers are not allowed to have any integrity algorithm other than "none") [2, 8].

c) *The Initial Exchanges*

Communication using IKE always begins with IKE_SA_INIT and IKE_AUTH exchanges (known in IKEv1 as Phase 1). These initial exchanges normally consist of four messages, though in some scenarios that number can grow. All communications using IKE consist of request/ response pairs. We'll describe the base exchange first, followed by variations. The first pair of messages (IKE_SA_INIT) negotiates cryptographic algorithms, exchange nonces, and do a Diffie-Hellman exchange [2].

The second pair of messages (IKE_AUTH) authenticate the previous messages, exchange identities and certificates and establish the first Child SA. Parts of these messages are encrypted and integrity protected with keys established through the IKE_SA_INIT exchange, so the identities are hidden from eavesdroppers and all fields in all the messages are authenticated. All messages following the initial exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the IKE_SA_INIT exchange [2].

All subsequent messages include an Encrypted payload, even if they are referred to in the text as "empty". For the CREATE_CHILD_SA, IKE_AUTH, or INFORMATIONAL exchanges, the message following the header is encrypted and the message including the header is integrity protected using the cryptographic algorithms negotiated for the IKE SA [2].

Every IKE message contains a Message ID as part of its fixed header. This Message ID is used to match up requests and responses and to identify retransmissions of messages [2].

In the following descriptions, the payloads contained in the message are indicated by names as listed below.

Notation	Payload
AUTH	Authentication
CERT	Certificate
CERTREQ	Certificate Request
HDR	IKE header (not a payload)
IDi	Identification - Initiator
IDr	Identification - Responder
KE	Key Exchange
Ni, Nr	Nonce
SA	Security Association
TSi	Traffic Selector – Initiator
TSr	Traffic Selector – Responder
V	Vendor ID

The initial exchanges are as follows:

```
Initiator                          Responder
-------------------------------------------------------------------
HDR, SAi1, KEi, Ni  -->
```

HDR contains the Security Parameter Indexes (SPIs), version numbers, and flags of various sorts. The SAi1 payload states the cryptographic algorithms the initiator supports for the IKE SA. The KE payload sends the initiator's Diffie-Hellman value. Ni is the initiator's nonce [2].

```
Initiator                          Responder
-------------------------------------------------------------------
<--  HDR, SAr1, KEr, Nr, [CERTREQ]
```

The responder chooses a cryptographic suite from the initiator's offered choices and expresses that choice in the SAr1 payload, completes the Diffie-Hellman exchange with the KEr payload, and sendsits nonce in the Nr payload [2].

d) The CREATE_CHILD_SA Exchange

The CREATE_CHILD_SA exchange is used to create new Child SAs and to rekey both IKE SAs and Child SAs. This exchange consists of a single request/response pair and some of its function was referred to as a Phase 2 exchange in IKEv1. It MAY be initiated by either end of the IKE SA after the initial exchanges are completed [2].

Either endpoint may initiate a CREATE_CHILD_SA exchange, so in this section the term initiator refers to the endpoint initiating this exchange [2]. If a CREATE_CHILD_SA exchange includes a KEi payload, at least one of the SA offers MUST include the Diffie-Hellman group of the KEi. The Diffie-Hellman group of the Kei MUST be an element of the group the initiator expects the responder to accept (additional Diffie-Hellman groups can be proposed). If the responder selects a proposal using a different Diffie-Hellman group (other than NONE), the responder MUST reject the request and indicate its preferred Diffie-Hellman group in the INVALID_KE_PAYLOAD Notify payload. There are two octets of data associated with this notification: the accepted Diffie-Hellman group number in big endian order. In the case of such a rejection, the CREATE_CHILD_SA exchange fails, and the initiator will probably retry the exchange with a Diffie-Hellman proposal and KEi in the group that the responder gave in the INVALID_KE_PAYLOAD Notify payload [2].

The responder sends a NO_ADDITIONAL_SAS notification to indicate that a CREATE_CHILD_SA request is unacceptable because the responder is unwilling to accept any more Child SAs on this IKE SA. This notification can also be used to reject IKE SA rekey. Some minimal implementations may only accept a single Child SA setup in the context of an initial IKE exchange and reject any subsequent attempts to add more [2].

i. *Creating New Child SAs with the CREATE_CHILD_SA Exchange*

A Child SA may be created by sending a CREATE_CHILD_SA request. The CREATE_CHILD_SA request for creating a new Child SA is:

```
Initiator                  Responder
-------------------------------------------------------------------
HDR, SK {SA, Ni, [KEi],
        TSi, TSr}  -->
```

The initiator sends SA offer(s) in the SA payload, a nonce in the Ni payload, optionally a Diffie-Hellman value in the Kei payload, and the proposed Traffic Selectors for the proposed Child SA in the TSiand TSr payloads [2].

The CREATE_CHILD_SA response for creating a new Child SA is:

```
Initiator                          Responder
-------------------------------------------------------------------
<--  HDR, SK {SA, Nr, [KEr],
                         TSi, TSr}
```

The responder replies (using the same Message ID to respond) with the accepted offer in an SA payload, and a Diffie-Hellman value in the KEr payload if KEi was included in the request and the selected cryptographic suite includes that group [2].

The Traffic Selectors for traffic to be sent on that SA are specified in the TS payloads in the response, which may be a subset of what the initiator of the Child SA proposed [2].

III. IKEv2 Mobility (MOBIKE)

IKEv2 is used for performing mutual authentication, as well as establishing and maintaining IPsec Security Associations (SAs) [2, 3]. In the base IKEv2 protocol [IKEv2], the IKE SAs and tunnel mode IPsec SAs are created implicitly between the IP addresses that are used when the IKE_SA is established. These IP addresses are then used as the outer (tunnel header) addresses for tunnel mode IPsec packets (transport mode IPsec SAs are beyond the scope of this document). Currently, it is not possible to change these addresses after the IKE_SA has been created [3].

There are scenarios where these IP addresses might change. One example is mobility: a host changes its point of network attachment and receives a new IP address [3]. Another example is a multi-homing host that would like to change to a different interface if, for instance, the currently used interface stops working for some reason.

The main scenario for MOBIKE is enabling a remote access VPN user to move from one address to another without reestablishing all security associations with the VPN gateway [3]. For instance, a user could start from fixed Ethernet in the office and then disconnect the laptop and move to the office's wireless LAN. When the user leaves the office, the laptop could start using General Packet Radio Service (GPRS); when the user arrives home, the laptop could switch to the home wireless LAN. MOBIKE updates only the outer (tunnel header) addresses of IPsec SAs, and the addresses and other traffic selectors used inside the tunnel stay unchanged. Thus, mobility can be(mostly) invisible to applications and their connections using the VPN [2, 3].

MOBIKE allows both parties to have several addresses and there are up to N*M pairs of IP addresses that could potentially be used. MOBIKE solves this problem by taking a simple approach: the party that initiated the IKE_SA (the "client" in a remote access VPN scenario) is responsible for deciding which address pair is used for the IPsec SAs and for collecting the information it needs to make this decision (such as determining which address pairs work or do not work). The other party (the "gateway" in a remote access VPN scenario) simply tells the initiator what addresses it has, but does not update the IPsec SAs until it receives a message from the initiator to do so.

Global Journal of Computer Science and Technology (E) Volume XIII Issue IX Version I Year 2013

This approach applies to the addresses in the IPsec SAs; in the IKE_SA case, the exchange initiator can decide which addresses are used [3].

A simple MOBIKE exchange in a mobile scenario is illustrated below.The notation is based on [IKEv2] [3].

Step 1 : Is the normal IKE_INIT exchange [2, 3].
Step 2 : The peers inform each other that they support MOBIKE [3].
Step 3 : The initiator notices a change in its own address and informs the responder about this by sending an INFORMATIONAL request containing the UPDATE_SA_ADDRESSES notification [3]. The request is sent using the new IP address. At this point, it also starts to use the new address as a source address in its own outgoing ESP traffic. Upon receiving the UPDATE_SA_ADDRESSES notification, the responder records the new address and, if it is required by policy, performs a return rout ability check of the address [3].

When this check (step 4) completes, the responder starts to use the new address as the destination for its outgoing ESP traffic.

a) Protocol Exchanges

i. *Initial IKE Exchange*

The initiator is responsible for finding a working pair of addresses so that the initial IKE exchange can be carried out. Any information from MOBIKE extensions will only be available later, when the exchange has progressed far enough. Exactly how the addresses used for the initial exchange are discovered is beyond the scope of this specification; typical sources of information include local configuration and DNS [3]. If either or both of the peers have multiple addresses, some combinations may not work. Thus, the initiator SHOULD try various source and destination address combinations when retransmitting the IKE_SA_INIT request [3].

ii. *Signaling Support for MOBIKE*

Implementations that wish to use MOBIKE for a particular IKE_SA MUST include a MOBIKE_SUPPORTED notification in the IKE_AUTH exchange (in case of multiple IKE_AUTH exchanges, in the message containing the SA payload) [3].

iii. *Initial Tunnel Header Addresses*

When an IPsec SA is created, the tunnel header IP addresses (and port, if doing UDP encapsulation) are taken from the IKE_SA, not the IP header of the IKEv2 message requesting the IPsec SA. The addresses in the IKE_SA are initialized from the IP header of the first IKE_AUTH request [3].

iv. *Additional Addresses*

Both the initiator and responder MAY include one or more ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS notifications in the IKE_AUTH exchange (in case of multiple IKE_AUTH exchanges, in the message containing the SA payload). Here "ADDITIONAL_*_ADDRESS" means either an ADDITIONAL_IP4_ADDRESS or an ADDITIONAL_IP 6_ADDRESS notification [3].

v. *Changing Addresses in IPsec SAs*

In MOBIKE, the initiator decides what addresses are used in the IPsec SAs. That is, the responder does not normally update any IPsec SAs without receiving an explicit UPDATE_SA_ADDRESSES request from the initiator. (As described below, the responder can, however, update the IKE_SA in some circumstances.) [3]

IV. Conclusions

The main goals of this research project are to maintain the security offered by usual IKEv2 procedures and to counter mobility-related threats in an appropriate manner. This section describes new security considerations introduced by MOBIKE.

1. Traffic Selector Authorization.
2. Traffic Redirection and Hijacking.
3. IPsec Payload Protection.
4. Denial-of-Service Attacks against Third Parties.
5. Spoofing Network Connectivity Indications.
6. Performance tuning for support to Mobile devices like Smartphones, IPADS, Tablets.

References Références Referencias

1. IKEv1, The Internet Key Exchange (IKE) RFC 2409, available athttp://tools.ietf.org/html/rfc2409
2. Internet Key Exchange Protocol Version 2 (IKEv2) IKEv2, RFC 5996, available at https://tools.ietf.org/html/rfc5996
3. IKEv2 Mobility and Multihoming Protocol (MOBIKE), http://tools.ietf.org/html/rfc4555
4. S. Frankel, et al. Guide to IPsec VPNs: Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-77, available at http://www.nist.gov
5. A. Uskov, "Information Security of Mobile VPN: Conceptual Models and Design Methodology", Proc. 2012 IEEE international conference on Electro/Information Technology EIT-2012. Indianapolis, IN; catalog number: CFP12EITCDR; ISBN: 978-1-4673-0818-2; ISSN: 2154-0373.
6. J. Carmouche, IPsec Virtual Private Network Fundamentals. Indianapolis, IN: Cisco Press, 2007.
7. V. Bollapragada, M. Khalid, S. Wainner, IPSec VPN Design. Indianapolis, IN: Cisco Press, 2005.
8. B. Schneider, Applied Cryptography: Protocols, Algorithms, and Source Code in C. Indianapolis, IN: Wiley, 1996.
9. N. Ferguson, B. Schneider, Practical Cryptography. Indianapolis, IN: Wiley, 2003.
10. N. Ferguson, B. Schneider, T. Kohno, Cryptography Engineering. Indianapolis, IN: Wiley, 2010.
11. IKEv2 Parameters, RFC4306, available at http://www.iana.org/assignments/ikev2-parameters
12. RSA Security, available at http://www.rsa.com