IPSec over Heterogeneous IPv4 and IPv6 Networks:Issues and Implementation

Nazrul M. Ahmad, and Asrul H. Yaacob

Faculty of Information, Science and Technology (FIST), Multimedia University (MMU), Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia

Summary

In the face of looming IPv4 address exhaustion and the slow pace of IPv4 to IPv6 migration, this work deploys the IPv4/IPv6 translation gateway as a mechanism to ensure most of IPv6 mission critical applications to continuously interoperate with legacy IPv4 nodes. However, the existence of translation gateway between two IPSec nodes from disparate address realms imposes some incompatibility issues due to the violation of TCP/UDP and IPSec intrinsic functionalities by the gateway. In this work, we study and explore the incompatibility issues of applying IPSec across the translation gateway and then propose a workable solution to implement end to end IPSec in heterogeneous IPv4 and IPv6 networks. Experimental results show that our mechanism is feasible to establish a successful IPSec connection across IPv4/IPv6 translation gateway. Moreover, the proposed mechanism provides automatic detection of IPv4/IPv6 translation gateway or Network Address Translation (NAT) gateway, so that it can be backward compatible with NAT-Traversal in homogeneous IPv4 networks.

Key words:

IP Security (IPSec); IPv4/IPv6 translation gateway; Internet Key Exchange (IKE); NAT-Traversal; End to end security

1. Introduction

The emergence of new Internet standard, Internet Protocol version 6 (IPv6) [1], is to mitigate the exhaustion of IPv4 address space and to enable a new evolution of IP networks and global communication. Unfortunately, IPv6 is not designed to provide backward compatibility with IPv4. Due to that, the migration from IPv4 to IPv6 may take longer than anticipated [2]. Hence, a number of transition tools have been developed to ensure the interoperability and coexistence of both IPv4 and IPv6 networks [3]. This enables the legacy IPv4 server applications to continuously interoperate with IPv6 clients for foreseeable future.

Although the transitions mechanisms provide interoperability between IPv6 and IPv4 nodes, but they impose some incompatibility issues when security features need to be added to the information to safeguard its transmission along untrusted medium. Authentication and authorization at the back-ends are not enough in case of there is possibility of security breaches in the each of intermediate device. Furthermore, to provide the protection of information from its origin to its destination, called end-to-end security, some encryption techniques are needed to be performed on the transmitting information. End to end security guarantees the safeguarding of the information along the communication path and not relies on the existence of the security facilities in intermediate devices.

IP Security (IPSec) is a suite of protocols developed to secure IP packets and it provides integrity, confidentiality, authentication and non-repudiation for the packets [4]. With the advent of IPv6, IPSec became a mandatory part of this new address space while it was optional in IPv4. Although most of the modern Operating Systems nowadays support IPSec for both IPv6 and IPv4 networks, but there are still so many issues and incompatibilities in applying IPSec between the nodes which are relied on different address realms and are located behind one or more translation gateway(s). In this study, we aim to dissect the issues arising when applying IPSec across translation gateway and then, propose a solution to establish IPSec connection between IPv6/IPv4 nodes. To provide clear elaboration of the proposed solution, we implement the proposed method by modifying the StrongSwan 4.5.0 [5] - a well-known IPSec software under Linux-based platform.

The structure of the paper is as follows: Section 2 briefly reviews the IPSec principles and Network Address Translation (NAT) concepts. This section also explores the incompatibility issues between translation gateway and IPSec. Section 3 extends the discussion by describing the complications of implementing IPSec over heterogeneous IPv4 and IPv6 networks. Then, Section 4 proposes a new IPv4/IPv6 translation gateway traversal mechanism to facilitate the establishment of end to end IPSec between IPv4 and IPv6 nodes. Section 5 elaborates on the development of Linux-based testbed to validate the proposed solution. Finally, Section 6 concludes this paper.

Manuscript received July 5, 2013
Manuscript revised July 20, 2013

2. Incompatibility Issues between IPSec and Translation Gateway

This section provides some background information about IPSec principles and describes common NAT issues.

2.1 Internet Protocol Security (IPSec)

IPSec is a protocol suite aims at securing end to end communications across the Internet. IPSec works in Internet layer of TCP/IP stack and provides confidentiality, integrity, authentication, and non-repudiation. There are two main transformations that are applied to an IP packet: Authentication Header (AH) [6] and Encapsulating Security Payload (ESP) [7]. AH provides connectionless integrity and data origin authentication. It also provides a service to protect against anti-reply attacks through a sequence number. AH header is inserted into the IP packet immediately after the outer IP header. Then, AH authenticates the entire packet including the preceding IP header. On the other hand, ESP may provide confidentiality, connectionless integrity, data origin authentication, and an anti-replay service. ESP header is inserted after the IP header and before the upper layer protocol header. Each of the ESP or AH protocols support two modes: transport mode and tunnel mode. The transport mode provides the end-to-end security service, while tunnel mode provides the security between security gateways, or between a security gateway and a host.

For automatic and secure negotiation of the IPSec security materials and options between two nodes, Internet Key Exchange (IKE) protocol is designed to facilitate the generation of shared cryptographic keying materials and to establish Security Association (SA) between the two communicating nodes [8]. SA is a set of policies including algorithms for authentication and encryption, and necessary parameters and also keys used for protecting the IKE negotiation and IP packets. IKE negotiation takes place in two phases. Phase 1 is the initial negotiation and an IKE SA is established, so that the end nodes agree on common proposal for securely exchanging IKE messages. In phase 2, another SA is established for IPSec communication.

2.2 Network Address Translation (NAT)

The depletion of IPv4 addresses causes insufficient design capacity of the original internet infrastructure. One of the short-term solutions to address the exhaustion was Network Address Translation (NAT), which allows multiple nodes to share one or more public IP addresses [9, 10]. A NAT gateway resides at the boundary of private and public networks and modifies private IP address and port of the packet that is destined to public network. IP packets which bundled with IPSec such as AH and ESP are intrinsically intended to protect the integrity of IP packet (including the source and destination addresses) from alteration or tampering. Since the fundamental role of NAT gateway is to modify the IP addresses in the packet’s header, IPSec and NAT has an intrinsic incompatibility. The following sections discuss on NAT and IPSec incompatibility issues and mention some solutions for the problems.

2.3 Traditional NAT Breaks IPSec

The co-existence of translation gateway and IPSec is not feasible due to several known incompatibilities [11]. The manipulation of IP header by NAT causes TCP/UDP checksum invalid and the IPSec integrity check to fail.TCP/UDP checksum has a dependency on IP source and destination addresses through the inclusion of TCP/UDP pseudo-header in the calculation. While the NAT gateway translates the IP addresses, it should also re-compute the checksum. Since IPSec ESP encrypts and authenticates ESP header and TCP/UDP header, any attempt to modify to checksum causes the IPSec integrity check to fail. Alternately, if NAT gateway does not update the checksum, TCP/UDP verification will fail. Thus, IPSec ESP only works if IPSec Tunnel mode is used or checksum calculation verification is omitted. Unlike IPSec ESP, IPSec AH includes IP header as part of IPSec integrity check calculation. When the NAT gateway modifies the IP header, IPSec evaluating this as violation of integrity and discards the packet. Therefore, AH and NAT gateway simply cannot work.

Another problem arises when IP address is used as identifier in IKE main mode negotiation using pre-shared key authentication. In this case, the translation of IP address results in discarding the packet [7]. Apart from those incompatibilities, if multiple peers located behind NAT initiate IPSec to the same node, then the overlapping Security Policy Database (SPD) may exist. This complication occurs when the peers use their source IP address as identifier [12]. The reason is that since NAT manipulates the peer IP address and translates it into maybe one public IP address, the node may send responses to the wrong peer due to equivalent existing SAs. On top of that, for NAT to multiplex IPSec connections for the peers, the translation of IKE UDP source port is mandatory. Since normal IKE traffics are exchanged on UDP port 500, the node now must be able to accept IKE traffic on UDP ports other than 500 and to reply to the same port.

2.4 Current NAT Traversal Solutions

Basically, NAT traversal is a mechanism for solving any NAT issue and for establishing and maintaining IP connections traversing NAT gateway. However, in this work, we address only IPSec-related NAT Traversal solutions.

Realm Specific IP (RSIP) is an alternative to NAT for handling end to end IPSec [13]. RSIP is designed based on Client/Server model. A RSIP server negotiates IP address translation parameters such as IP address and port with RSIP client. Based on these information, RSIP client prepares the packet that are ready for the public network or initiates end to end IPSec negotiation with remote end, such that no translation is necessary by the RSIP server. The RSIP approach does not require the modification to the IKE negotiation, but the modification TCP/IP stack. On top of that, RSIP requires the changes on existing physical structure of network and quite costly or not universally applicable for short-term.

Traversal Using Relay NAT (TURN) is another variant of NAT traversal solutions [14]. This mechanism requires a dedicated TURN server to communicate with the node behind a NAT (called TURN client). A TURN client sends a request to a TURN server to obtain a transport address (IP address and port number) prior to establishing the connection. Based on the allocated transport address, the TURN server is responsible to relay traffic between the TURN client and its peer. TCP transport address allocated by TURN server properly works with TLS and SSL. However, any addresses allocated by TURN server will not operate properly with IPSec AH in transport mode. IPSec ESP and any tunnel-mode ESP or AH should still operate [25].

NAT-Traversal (NAT-T) [15] offers an effective and simple way to support IPSec across the translation gateway. Figure 1 shows NAT-T operation in the IKE main mode negotiation by using digital signature authentication. The first step of NAT-T operation is to detect the NAT-T capability of remote nodes. This is done by exchanging special Vendor ID (VID) in the first two messages of the IKE Phase 1 negotiation. Successful exchange of VID indicates that both ends support NAT-T. Upon agreeing both ends support NAT-T, the second step is to discover the presence of translation gateways along the communication path. The presence of translation gateway is determined by sending NAT-D (NAT-Discovery) payload. Both ends calculate and send the hashes of IP addresses and ports of the source and destination nodes. The NAT-D payloads are exchanged in the third and fourth messages of the main mode. Once received the NAT-D payload, both ends recalculate the hash value based on the IP and port addresses that they used in the IKE negotiation. If these hashes do not match, both ends discover that translation gateway exists somewhere in between.

The packet header translation by the translation gateway causes TCP/UDP checksum to fail. To validate the checksum, both end nodes need to exchange their original addresses used to construct the packets. The original addresses are embedded into a NAT-OA (Original Address) payload. This gives responder node accesses to required information so that the source and destination IP addresses can be checked and the checksum can be validated. The NAT-OA payloads are exchanged during the first and second messages of IKE Phase 2 as shown in Figure 2. On receiving NAT-OA payloads, the end nodes assemble the TCP/UDP pseudo-header based on the information contained in the NAT-OA payloads. This pseudo-header is appended to the original TCP/UDP segment for recalculating the checksum.

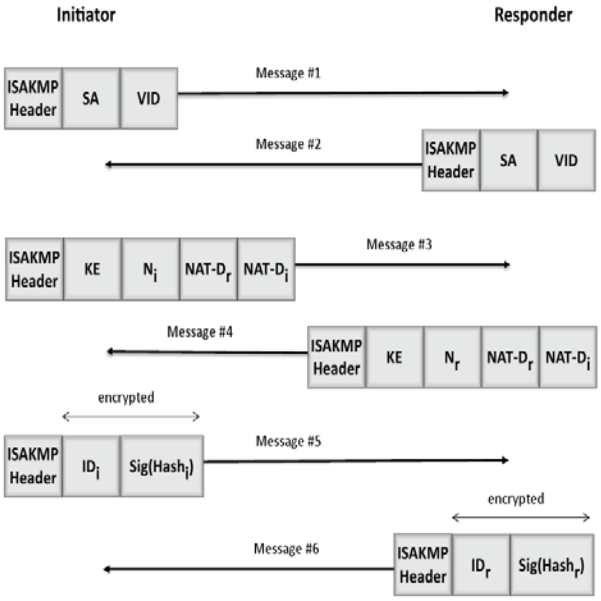


Figure 1: IKE Main Mode with NAT Detection

With NAT-T, the UDP-ESP encapsulation [16] is deployed to support the IPSec packets from both end nodes to traverse the translation gateway and to avoid any problems with the IPSec-aware translation gateway. Let's imagine multiple connections are mapped to one allocated address. Since IPSec ESP does not use port information, the translation gateway can only utilize the protocol field in IP header to distinguish the packets. When the first IPSec connection is established, the translation gateway maintains the translation state information in the table so that all IPSec ESP packets will be routed to the first connection. However, when there is a new IPSec connection, the translation gateway replaces the entry in the table and thus breaking the first IPSec connection. UDP-ESP encapsulation gives the translation gateway an UDP header containing UDP port that can be used for multiplexing IPSec data streams.

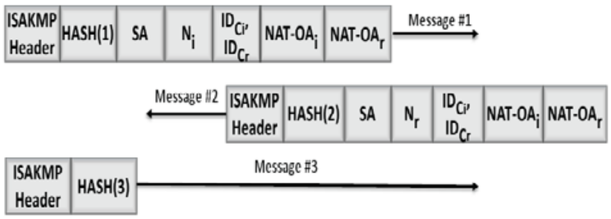


Figure 2: IKE Quick Mode with NAT-OA Payload

3. IPSec across IPv4/IPv6 Translation Gateway

IPv6 is the extension of current Internet Protocol version 4 (IPv4) and it is developed to solve the address exhaustion of its prior [1]. But, unfortunately it does not provide backward compatibility with IPv4. As a result, some transition mechanisms are needed when heterogeneous networks wants to communicate. The transition mechanisms which proposed to solve the incompatibility problems between IPv4/IPv6 are dual stacks [17], tunnelling [18] and translation [19]. The most common transition mechanism nowadays is dual stack. A node with a dual stack allows co-existence and interoperability of IPv4 and IPv6 nodes using IPv4 or IPv6 packets. This permits gradual application-by-application upgrades into IPv6 environment without much disruption. In IPv6 transition, tunneling is basically used to enable one network to send its data via another network's connections. Thus, 6to4 tunneling is done by encapsulating IPv6 packets within IPv4 packets and sends them as native IPv4 traffic. The tunnel provides virtual links over IPv4 physical network. When IPv4 and IPv6 nodes are not directly compatible and when IPv6 node wants to communicate with numerous IPv4 nodes, translation mechanism should be used. The translation mechanism such as Network Address Translation – Protocol Translation (NAT-PT) was proposed without any modification to nodes on both ends whereby the router or translation gateway takes the responsibility of the translation.

NAT-PT is a mechanism for allowing native IPv6 to communicate with native IPv4 and vice versa [19]. A NAT-PT is located at the boundary of IPv6 and IPv4 address realms. Each NAT-PT gateway retains a pool of globally routable IPv4 address and an IPv6 prefix with a prefix length of 96 bits. The IPv6 prefix can be a unique local unicast prefix or can be any globally routable prefix. IPv4 address pool is used to assign to IPv6 nodes on a dynamic basis as sessions are initiated across the NAT-PT gateway. On the other hand, IPv6 prefix is static and any IP addresses from packets originating from IPv4 nodes destined to the IPv6 network will be translated and formed IPv4-mapped NAT-PT IPv6 addresses. For stateless translation, the IPv4 address is used as the lower 32 bits of the IPv6 address and is appended to the IPv6 prefix. All packets originating in IPv6 network use this IPv4-mapped NAT-PT IPv6 addressing convention as part of the IPv6 destination addresses and packets carrying that prefix are routed to the NAT-PT gateway.

3.1 IPSec Issues when Applying to NAT-PT

NAT-PT inherits NAT incompatibilities and problems when applying IPSec across the NAT-PT gateway. It breaks TCP/IP intrinsic end to end functionalities and hence, any IP-based applications protected by IPSec cannot traverse NAT-PT gateway [20]. The major problems of applying IPSec across disparate address realms can be categorized as follows. First, similar to traditional NAT, the presence of NAT-PT gateway violates the TCP/UDP checksum and fails to verify the integrity check value of IPSec due to the change of the IP header. Any attempts to make TCP/UDP checksum verifiable at the receiver node should be designed carefully due to the existence of disparate address realms. In case of applying existing NAT-T to resolve this problem over IPSec ESP, TCP/UDP checksum re-computation relies on NAT-OA payload. If the IPv6 packet traverses across NAT-PT gateway, the gateway modifies the outer IPv6 header into IPv4 header. But the NAT-OA inside the payload is not changed because of ESP packet is encrypted. Upon receipt of the packet, IPv4 node recalculates to verify TCP/UDP checksum using NAT-OA payload. Since, IPv4 node has only native IPv4 network protocol stack, so it cannot parse the NAT-OA.

Secondly, the disparate address realms cause the IKE negotiation to be failed due to mismatch of Identification (ID) payload type in IKE phase 1 negotiation [21]. The node has wide variety of ID payload type to be chosen as a node's identity [8]. The problem arises when the node chooses IP-based ID payload type as its identity. For instance, IPv6 node could select one of these ID payload types: ID_IPV6_ADDR, ID_IPV6_ADDR_SUBNET, and ID_IPV6_ADDR_RANGE. Since the ID payloads are sent during the third IKE exchange and are encrypted, identities will not be translated by the NAT-PT gateway. IPv4 node which has only native IPv4 protocol stack is expected by default to set and to receive the identity from the same IP address family.

Finally, IPSec across heterogeneous networks is failed to establish because of a mismatch in local SA policy and quick mode identities. The identities of the IPSec SAs negotiated in quick mode are implicitly assumed to be the IP addresses of the nodes, without any implied constraints on the protocols or post numbers allowed, unless specific identities are specified in quick mode [8]. Both initiator and responder nodes use identities in quick mode ID payloads as selectors to search the local SA policy that has been established prior to IPSec SA negotiation for matching connection. Since the attributes in local SA policy is set based on transmitted IKE messages, the acceptable source and destination IP addresses of the local SA policy must match the negotiated quick mode identities. Local policy will dictate whether the IPSec SA request proposals are acceptable for the identities specified. As identities in quick mode can be only based on IP-based ID payload type, the node specifies the ID_{ci} and ID_{cr} payloads with the IP address, range or subnet from its address realm perspective. Mismatch occurs when source and destination addresses of the IKE messages are translated by NAT-PT gateway.

3.2 Related Work

Several approaches for addressing the limitations of applying IPSec across IPv4/IPv6 translation gateway have been proposed, most of which aimed to solve the failure of TCP/UDP checksum and IPSec integrity validation.

Souhwan et al. proposed an approach to use IP Header Translation Information (IP-HTI) message between the IPSec initiator and the NAT-PT gateway [22]. The message which includes the allocated IPv4 address for IPv6 IPSec initiator and IPv6 prefix information of the NAT-PT gateway is sent in advance to the IPSec initiator during IKE negotiation process. The purpose behind these are simply to allow the IPSec initiator to calculate the right TCP/UDP checksum for ESP packets and to generate the correct $HASH_i$ for IKE message 5, and make them verifiable at the responder.

There is inevitable problem to the IP-HTI approach. The approach has limitation to establish multiple IPSec communications among many IPv6 nodes and IPv4 nodes only through one globally routable IPv4 address. Peng et al. proposed almost identical concept to IP-HTI, i.e., IP Translation message (IP_TI) [20]. IP_TI message includes all information defined in IP-HTI with the addition of 16-bit port translation parameters. This approach modifies address mapping table and session table during IKE negotiation process to establish multiple bidirectional sessions among IPv6 and IPv4 nodes through one globally registered routable IPv4 address. However, both approaches require the IKE/IPSec-aware NAT-PT gateway in order to generate IP-HTI and IP_TI messages.

In [23], a new NAT-PT/IPSec traversal approach was proposed, namely T-NATPT. The approach is based on the modification of existing NAT-T solution. This approach also adopts UDP-ESP encapsulation to support the IPSec packets from both end nodes to traverse across NAT-PT gateway and to avoid any problems with the IKE/IPSec-aware NAT-PT gateway. There are two important features are introduced in the IKE main mode, i.e., the detection of T-NATPT capability support and the detection of NAT-PT along the communication path. T-NATPT capability for peers is determined by the exchange of special Vendor ID. For NAT-PT detection, a Translation Discovery (TD) payload is sent during the third exchange of IKE main mode. Unlike the NAT-D payload in NAT-T, TD payload includes extra parameter on IP protocol used by the node. Once received TD payload, the node recalculates the hash values and compares with the negotiated hash values inside the TD payload. If the values are matched, both nodes agree that no NAT-PT or NAT between them. Otherwise, both nodes need to examine the protocol field to determine whether NAT-PT or NAT exists along the communication path. If protocol field is not matched with the node's IP address realm, then NAT-PT is detected. To facilitate TCP/UDP checksum fixing at both nodes, IPSec Relation Announcement (IRA) payloads are exchanged during IKE quick mode.

However, there are few questionable steps need to be justified in T-NATPT approach. First, the approach encapsulates all IKE messages with UDP-non-ESP encapsulation even when the NAT-PT has not yet been detected. Unnecessary encapsulation of the IKE message at the beginning of the negotiation induces extra processing at the node and does not provide backward compatibility with normal IKE negotiation and to NAT-T. In existing NAT-T, IKE messages only will be encapsulated when the NAT is detected along the communication path. This is done normally after the second exchange of IKE messages. In contrast to normal IKE negotiation, no UDP-non-ESP encapsulation is required. Secondly, the approach also proposes the exchange of encrypted TD payload in third exchange of IKE negotiation. Since the TD payload just contains hash values of the IP addresses and port numbers, and IP protocol, no vital information is disclosed in this payload. Unlike NAT-T, the NAT-D payloads are exchanged in the second exchange of IKE messages and are not encrypted.

4. E2E-NATPT Traversal

The emergence of IPv6 and the slow pace of IPv4 to IPv6 migration should not adversely impact on the coexistence of the heterogeneous IPv4 and IPv6 networks. To coexist, the deployment of IPv4/IPv6 translation gateway provides seamless translations for IPv4 and IPv6 interoperability and offers better scalability. However, as discussed in the previous sessions, the presence of translation gateway causes some incompatibilities issues between the gateway and IPSec. Adopting NAT-T and UDP-ESP encapsulation methods, we propose a new mechanism to address the establishment of IPSec across the IPv4/IPv6 translation gateway. Apart from solving the failure of TCP/UDP checksum and IPSec integrity validation, the mechanism called E2E-NATPT is proposed to address all issues arise during IKE negotiation between two nodes from disparate address realms. E2E-NATPT provides solution for end to end IPSec between two end nodes across the NAT-PT gateway. In this matter, we deploy ESP transport mode since it is feasible for end to end security. This session discusses in detail about our proposed solution.

4.1 IKE Main Mode

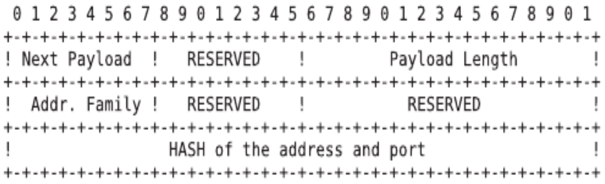
4.1.1 Detecting Support of E2E-NATPT Traversal

To initiate IPSec across NAT-PT gateway, E2E-NATPT traversal capability of the remote node is determined by the exchange of special Vendor ID (VID) payload. During first two messages of IKE main mode, initiator proposes a list of VIDs and the responder can only send the corresponding VIDs back to the initiator if it supports them. Thus, once E2E-NATPT VID payloads successfully exchanged, the nodes are acknowledged on their capability to support E2E-NATPT and are triggered to detect the presence of NAT-PT gateway between the nodes in the second exchange of IKE messages. To proclaim the capability of the nodes to support E2E-NATPT, the end nodes exchange MD5 hash of “E2E-NATPT”. The content of the VID payload in hexadecimal as follows:

“47BA7CC21A4F25F9CDC8BD1414F9F79B”

4.1.2 Detecting the presence of NAT-PT Gateway

For detection of NAT-PT gateway, we inherit the similar concept in NAT-T. However, NAT-D payload is not capable to provide sufficient information to determine the existence of disparate address realms between the two nodes. Besides sending the hashes of IP addresses and ports of the nodes, we enhance NAT-D payload and derive a new NATPT-D (NATPT-Discovery) payload as shown in Figure 3. We propose to include the IP address family type of the communicating node in the NATPT-D payload. The field contains the AF_INET value or AF_INET6 value if the source or destination IP address of the node is IPv4 or IPv6 respectively. The detection of NAT-PT gateway is accomplished in two stages. In the first stage, NATPT-D payloads are used to detect the presence of any translation gateway between the two nodes. The mismatch between the transmitted hashes and recalculated hashes indicates the presence of the gateway. The second stage of NAT-PT detection is to determine the existence of disparate address realms between the two nodes. The address family field inside the NATPT-D gives the indicator to the receiving node on the address realm of the other node. If both nodes are originated from disparate address realms, then NAT-PT gateway exists between them. Otherwise, it just normal NAT gateway. The proposed mechanism provides automatic detection of NAT-PT gateway or NAT gateway, so that it can be backward compatible with NAT-Traversal in homogeneous IPv4 networks.



```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !   RESERVED    !        Payload Length         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Addr. Family !   RESERVED    !          RESERVED             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                 HASH of the address and port                  !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: NATPT-Discovery Payload

4.1.3 Selection of Identification Payload Type in IKE Third Exchange

As mentioned in sections 2 and 3, the use of IP-based ID payload type as a node’s identifier causes IKE authentication to fail due to invalid ID information. The presence of NAT-PT gateway restricts the choices of IKE identifier to ID_FQDN and ID_USER_FQDN. While an ID type of ID_FQDN can be selected for machine authentication, ID_USER_FQDN is more suitable for user authentication [14].

4.2 IKE Quick Mode

4.2.1 Matching of Local SA Policy and Traffic Selectors

IKE SA is established once the IKE main mode is successfully negotiated between two end nodes. The ensuing Phase 2 IKE quick mode uses three additional messages to exchange the traffic selectors and cryptographic transforms needed for an IPSec SA. Hence, the matching of the negotiated traffic selectors and the established local IKE SA policy is one of the deciding factors in establishment of IPSec SA. From previous discussion, the presence of NAT-PT gateway causes the mismatch between them. Figure 4 illustrates the extent of the problem. Responder’s local IKE SA is established based on the transmitted IKE messages. Since the IPv6 addresses of the transmitted IKE messages are translated by NAT-PT gateway, local IKE SA is set up based on the translated IP addresses. At the beginning of the IKE quick mode, the initiator selects allowable identifiers as traffic selectors. Theses traffic selectors will be used by the responder to choose on the agreed local SA policy. Considering only IP-based identifiers are permitted, the initiator initializes the ID_{ci} and ID_{cr} payloads based on its address realm perspective. That means, for end to end connection, ID_{ci} and ID_{cr} payloads can only be initialized with ID payload type ID_IPV6_ADDR. Therefore, mismatch occurs when IPv6 traffic selectors are used to match the IPv4-based local SA policy.

To solve the problem, we propose to make modification to IKE daemon of IPSec protocol stack. Since the initiator is impossible to select appropriate traffic selectors without knowing the NAT-PT translation information such as IPv4 address pool and NAT-PT IPv6 prefix and there is no interaction between the initiator and NAT-PT gateway, we propose a method to manipulate the transmitted traffic selectors for matching purpose. In this case, responder's IKE daemon manipulates the transmitted IPv6 traffic selectors into IPv4 traffic selectors while retaining vital information about port numbers and protocol.

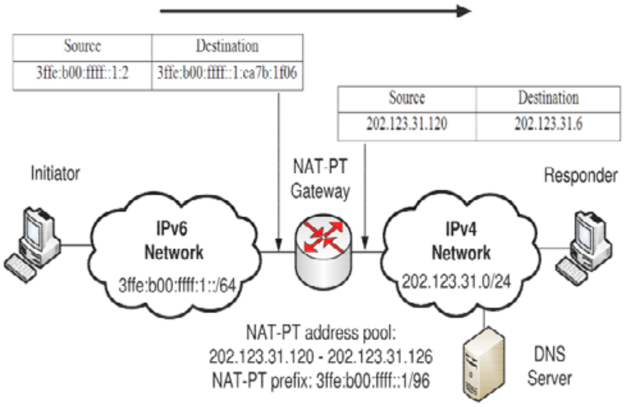


Figure 4: E2E-NATPT Implementation

First of all, we address the traffic selector in ID_{ci} payload. This payload holds the initiator's IP address namely 3ffe:b00:ffff:1::1. In this implementation, we assume the responder acts as a server and it capable to accept any incoming connection. Since the initiator has been authenticated in the IKE Main Mode and routed connection of the IKE has been identified by the initiator and responder cookies, responder's IKE daemon will change the traffic selector to 0.0.0.0 or also known as no IP address. Since the initiator has no information about the translation process by the NAT-PT gateway, the initiator fails to choose best traffic selector to match the exact local IKE SA policy. Therefore, the responder abstracts the traffic selector to be no IP address in order to match it against a list of possible opportunistic connections of the local SA policy.

Alternately, ID_{cr} payload specifies the responder's IP address namely 3ffe:b00:ffff::1:ca7b:1f06. This address represents the translated IPv4 address of the responder. Generally, this address is formed by simply concatenating NAT-PT IPv6 prefix with the 4-octet IPv4 address of the node. Since this implementation assumes the prefix-length of the NAT-PT prefix is always 96 bits, the responder's IKE daemon will extract the lower 32 bits from the perceived IPv6 address to obtain the IPv4 address of the responder. As a result, both transformed traffic selectors now can be used to check against the local IKE SA policy.

5. Implementation

5.1 Testbed

As a preliminary proof of concept, we built this testbed under Linux platform. The IPv4/IPv6 internetworking environment for end to end IPSec support is illustrated in Figure 4. Our testbed consists of personal computers running on Ubuntu 10.04 LTS with Linux kernel version 2.6.32. The initiator is an IPv6-only client and the responder is an IPv4-only legacy Apache-based HTTP server [25]. A translation gateway with NAT-PT capability is responsible for translating IPv6 packets to IPv4 packets, and vice versa. To facilitate IPv4 and IPv6 name to IP address mapping, BIND DNS server is used [26]. Since the nodes are distributed across the disparate address realms, DNS queries and responses need to cross the translation gateway. Therefore, DNS Application Level Gateway (ALG), an application specific agent that translates IPv6 addresses in DNS queries and responses into their IPv4 address bindings, is used in conjunction with NAT-PT to provide support for such application. To accomplish this task, Cisco router 2811 with IOS Advanced IP Services is deployed.

To implement end to end IPSec across heterogeneous IPv4 and IPv6 networks, we use StrongSwan [5], an open source IPSec-based VPN solution for Linux. StrongSwan is chosen since it more stable release software after the preceding OpenSwan [27] and obsolete FreeSwan. StrongSwan has two major components, Pluto IKEv1 Daemon and Kernel IPSec Support (NETKEY). Pluto IKE Daemon is an IPSec implementation of IKE negotiation and authentication. To support E2E-NATPT traversal on StrongSwan, we modify Pluto IKEv1 daemon to enhance existing NAT-T capability. Partial IPSec configuration of /etc/ipsec.conf for both initiator and responder are shown in Table 1. In this implementation, we design in such a way the server is capable to accept multiple IPSec connections.

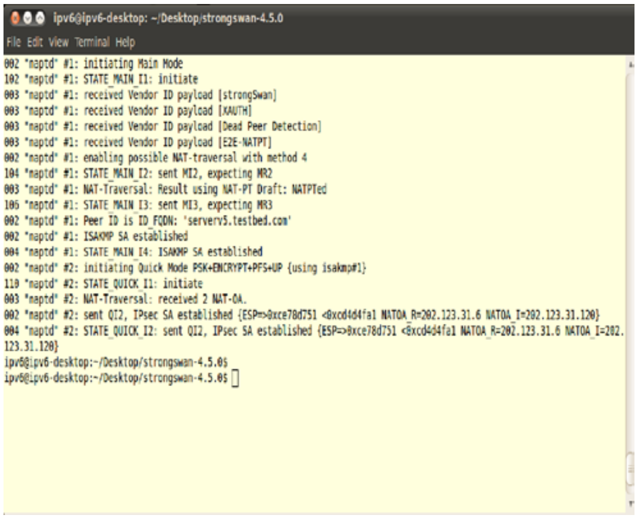
Table 1: Partial IPSec configuration in ipsec.conf

Initiator	Responder
Left=3ffe:b00:ffff:1::2	Left=serverv5.testbed.com
Leftid=@ipv6.testbed.com	Leftid=@serverv5.testbed.com
Right=serverv5.testbed.com	Right=%any
Rightid=@serverv5.testbed.com	Rightid=%any
Type=transport	Type=transport
Keyexchange=ikev1	Keyexchange=ikev1

5.2 Results and Discussion

The detection of translation gateway during the second exchange of IKE negotiation forces the latter stages of negotiation and communication to be encapsulated with UDP packet at port 4500. As mentioned earlier, the UDP encapsulation is primarily designed for IPSec to traverse the IPSec-aware NAT gateway in public-private IPv4 homogeneous network and to address many incompatibility issues between IPSec and the gateway. The UDP encapsulation process occurs at two stages, software level and OS kernel level. In our context, the encapsulation of IKE negotiation is performed by StrongSwan (i.e. software level). On the other hand, the encapsulation of IPSec ESP packet is executed at Linux kernel level.

During the negotiation of IKE quick mode, StrongSwan initializes the agreed SA policy into Linux kernel to monitor and to secure IP packets for the end to end connection. The Linux kernel creates an XFRM state and stores it in the memory. This is as a reference on how the IPSec connection with a particular Security Parameter Index (SPI) value to be handled. The state includes the algorithms and the keys for authentication and encryption used in ESP connection. The state also specifies the requirement of UDP encapsulation to ESP packet due to the presence of translation gateway. The initialization of XFRM state is completely succeeded in the IPv4 address realm. However, the initialization of XFRM state in the IPv6 node is failed and the negotiation is ended with incomplete connection. The failure of IPSec establishment is caused by the missing of UDP encapsulation functionality in the IPv6 stack of Linux kernel.


ipv6@ipv6-desktop: ~/Desktop/strongswan-4.5.0
File Edit View Terminal Help

```
002 "naptd" #1: initiating Main Mode
102 "naptd" #1: STATE_MAIN_I1: initiate
003 "naptd" #1: received Vendor ID payload [strongSwan]
003 "naptd" #1: received Vendor ID payload [XAUTH]
003 "naptd" #1: received Vendor ID payload [Dead Peer Detection]
003 "naptd" #1: received Vendor ID payload [E2E-NATPT]
002 "naptd" #1: enabling possible NAT-traversal with method 4
104 "naptd" #1: STATE_MAIN_I2: sent MI2, expecting MR2
003 "naptd" #1: NAT-Traversal: Result using NAT-PT Draft: NATPTed
106 "naptd" #1: STATE_MAIN_I3: sent MI3, expecting MR3
002 "naptd" #1: Peer ID is ID_FQDN: 'serverv5.testbed.com'
002 "naptd" #1: ISAKMP SA established
004 "naptd" #1: STATE_MAIN_I4: ISAKMP SA established
002 "naptd" #2: initiating Quick Mode PSK+ENCRYPT+PFS+UP {using isakmp#1}
110 "naptd" #2: STATE_QUICK_I1: initiate
003 "naptd" #2: NAT-Traversal: received 2 NAT-OA.
002 "naptd" #2: sent QI2, IPsec SA established {ESP=>0xce78d751 <0xcd4d4fa1 NATOA_R=202.123.31.6 NATOA_I=202.123.31.120}
004 "naptd" #2: STATE_QUICK_I2: sent QI2, IPsec SA established {ESP=>0xce78d751 <0xcd4d4fa1 NATOA_R=202.123.31.6 NATOA_I=202.
123.31.120}
ipv6@ipv6-desktop:~/Desktop/strongswan-4.5.0$
ipv6@ipv6-desktop:~/Desktop/strongswan-4.5.0$ 
```

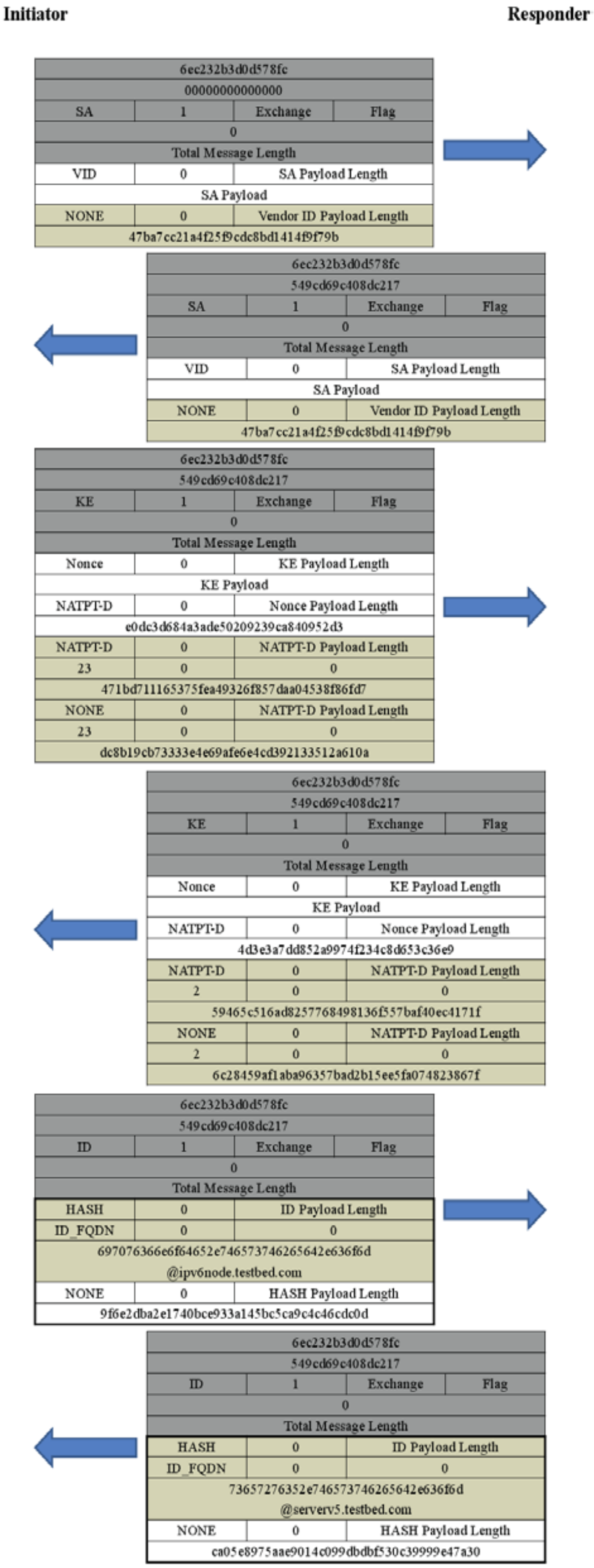
Figure 5: Capture of IKE negotiation in Ubuntu terminal

Since the UDP encapsulation is designed as a workaround against IPSec over NAT, there is no reason to have the same functionality in IPv6 protocol. The IPv6 protocol eliminates the need of native IPv6 to IPv6 NAT due to vast amount of IP addresses that can uniquely identify the nodes on the Internet and allow true end to end communication [28]. Nevertheless, the slow migration from IPv4 to IPv6 and the necessity to deploy NAT-PT gateway as a temporary solution to address the incompatibility between both address realms, require the UDP encapsulation to be adapted once again to solve the limitation. In this work, we write a simple patch to IPv6 stack by instantiating the UDP encapsulation concept used in IPv4 stack. With the introduction of the patch, the IPSec negotiation and communication between IPv4 and IPv6 nodes can be established successfully.

Next, we demonstrate the IKE negotiation with E2E-NATPT capability as an IPSec connection is established between initiator and responder. Figure 5 is the capture of initiator's Linux terminal when IPv6 client initiates IPSec connection to IPv4 HTTP web server. For an in-depth analysis of IKE negotiation, Figures 6 and 7 show the exchange of IKE payloads and the content of the payloads.

As described in the previous section, the first and second IKE main mode exchanges are used to detect the support of E2E-NATPT traversal between the two nodes and to detect the presence of NAT-PT gateway between them. E2E-NATPT VID payload and NATPT-D payload are sent to accomplish these two mechanisms. In particular, the address family field in NATPT-D payload shows the origin address realm of IKE message. In this case, two possible values are used: (2) for AF_INET and (23) for AF_INET6. Due to the constraint on the selection of ID payload type, we choose ID_FQDN as the identifier for initiator and responder. The ID payloads are illustrated in the third exchange of the IKE main mode. The thick border of the IKE messages indicates that the payloads are encrypted.

Our proposed solution, as explained earlier, does not change any IKE quick mode message. This can be shown in Figure 7. The traffic selectors ID_{ci} and ID_{cr} are set based on the source and destination IP addresses of the initiator. However, once NAT-PT gateway is detected by both nodes, the responder's IKE daemon modifies the transmitted traffic selectors for matching with local SA policy. Otherwise, the IKE daemon will bypass the procedure. During IKE quick mode negotiation, we can also observe the exchange of NAT-OA payloads for TCP/UDP checksum validation. In this matter, since StrongSwan not fully supports the exchange of double NAT-OA payloads (original source and destination IP addresses), we enhance StrongSwan to accomplish the exchange. The exchange of double NAT-OA payloads is depicted in Figure 5 and Figure 7.



6ec232b3d0d578fc			
00000000000000			
SA	1	Exchange	Flag
0			
Total Message Length			
VID	0	SA Payload Length	
SA Payload			
NONE	0	Vendor ID Payload Length	
47ba7cc21a4f25f9cdc8bd1414f9f79b			

6ec232b3d0d578fc			
549cd69c408dc217			
SA	1	Exchange	Flag
0			
Total Message Length			
VID	0	SA Payload Length	
SA Payload			
NONE	0	Vendor ID Payload Length	
47ba7cc21a4f25f9cdc8bd1414f9f79b			

6ec232b3d0d578fc			
549cd69c408dc217			
KE	1	Exchange	Flag
0			
Total Message Length			
Nonce	0	KE Payload Length	
KE Payload			
NATPT-D	0	Nonce Payload Length	
e0dc3d684a3ade50209239ca840952d3			
NATPT-D	0	NATPT-D Payload Length	
23	0	0	
471bd711165375fea49326f857daa04538f86fd7			
NONE	0	NATPT-D Payload Length	
23	0	0	
dc8b19cb73333e4e69afe6e4cd392133512a610a			

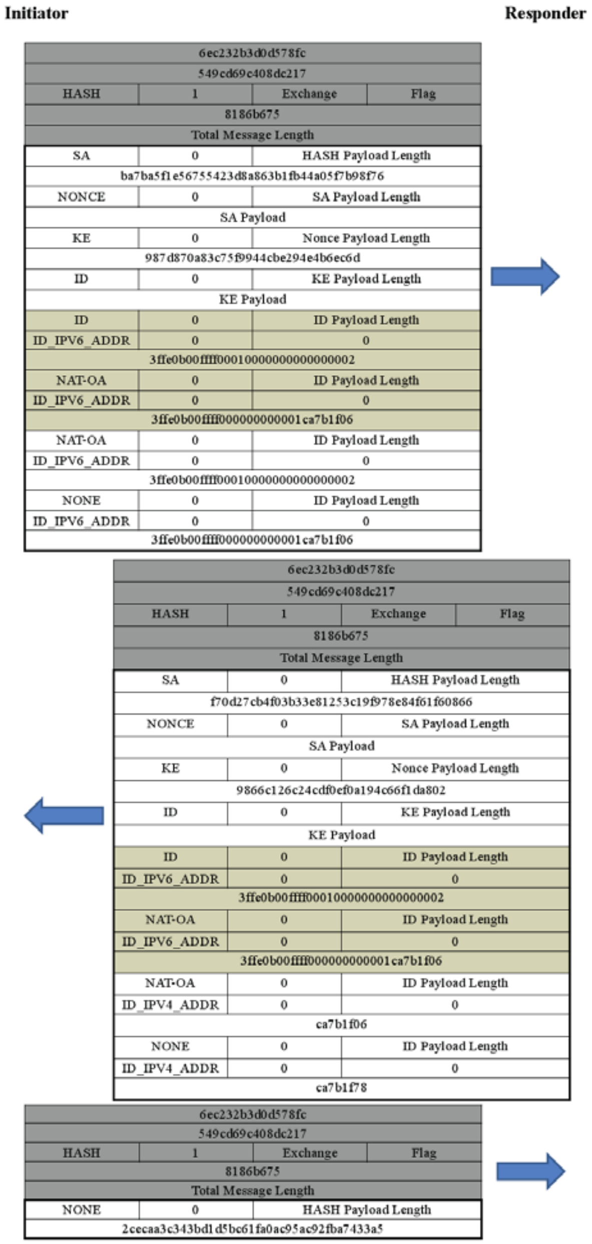
6ec232b3d0d578fc			
549cd69c408dc217			
KE	1	Exchange	Flag
0			
Total Message Length			
Nonce	0	KE Payload Length	
KE Payload			
NATPT-D	0	Nonce Payload Length	
4d3e3a7dd852a9974f234c8d653c36e9			
NATPT-D	0	NATPT-D Payload Length	
2	0	0	
59465c516ad8257768498136f557baf40ec4171f			
NONE	0	NATPT-D Payload Length	
2	0	0	
6c28459af1aba96357bad2b15ee5fa074823867f			

6ec232b3d0d578fc			
549cd69c408dc217			
ID	1	Exchange	Flag
0			
Total Message Length			
HASH	0	ID Payload Length	
ID_FQDN	0	0	
697076366e6f64652e746573746265642e636f6d @ipv6node.testbed.com			
NONE	0	HASH Payload Length	
9f6e2dba2e1740bce933a145bc5ca9c4c46cdc0d			

6ec232b3d0d578fc			
549cd69c408dc217			
ID	1	Exchange	Flag
0			
Total Message Length			
HASH	0	ID Payload Length	
ID_FQDN	0	0	
73657276352e746573746265642e636f6d @serverv5.testbed.com			
NONE	0	HASH Payload Length	
ca05e8975aae9014c099dbdbf530c39999e47a30			

Figure 6: The negotiations of IKE main mode messages

6. Conclusion



6ec232b3d0d578fc			
549cd69c408dc217			
HASH	1	Exchange	Flag
8186b675			
Total Message Length			
SA	0	HASH Payload Length	
ba7ba5f1e56755423d8a863b1fb44a05f7b98f76			
NONCE	0	SA Payload Length	
SA Payload			
KE	0	Nonce Payload Length	
987d870a83c75f9944cbe294e4b6ec6d			
ID	0	KE Payload Length	
KE Payload			
ID	0	ID Payload Length	
ID_IPV6_ADDR	0	0	
3ffe0b00ffff00010000000000000002			
NAT-OA	0	ID Payload Length	
ID_IPV6_ADDR	0	0	
3ffe0b00ffff000000000001ca7b1f06			
NAT-OA	0	ID Payload Length	
ID_IPV6_ADDR	0	0	
3ffe0b00ffff00010000000000000002			
NONE	0	ID Payload Length	
ID_IPV6_ADDR	0	0	
3ffe0b00ffff000000000001ca7b1f06			

6ec232b3d0d578fc			
549cd69c408dc217			
HASH	1	Exchange	Flag
8186b675			
Total Message Length			
SA	0	HASH Payload Length	
f70d27cb4f03b33e81253c19f978e84f61f60866			
NONCE	0	SA Payload Length	
SA Payload			
KE	0	Nonce Payload Length	
9866c126c24cdf0ef0a194c66f1da802			
ID	0	KE Payload Length	
KE Payload			
ID	0	ID Payload Length	
ID_IPV6_ADDR	0	0	
3ffe0b00ffff00010000000000000002			
NAT-OA	0	ID Payload Length	
ID_IPV6_ADDR	0	0	
3ffe0b00ffff000000000001ca7b1f06			
NAT-OA	0	ID Payload Length	
ID_IPV4_ADDR	0	0	
ca7b1f06			
NONE	0	ID Payload Length	
ID_IPV4_ADDR	0	0	
ca7b1f78			

6ec232b3d0d578fc			
549cd69c408dc217			
HASH	1	Exchange	Flag
8186b675			
Total Message Length			
NONE	0	HASH Payload Length	
2cecaa3c343bd1d5bc61fa0ac95ac92fba7433a5			

Figure 7: The negotiations of IKE quick mode messages

This work focused on describing and exploring the incompatibilities issues between IPSec and IPv4/IPv6 translation gateway and targeted to solve the issues by proposing NAT-PT traversal mechanism called E2E-NATPT. E2E-NATPT is responsible to support and guarantee end-to-end traversal of IPSec packets between IPv6 and IPv4 nodes. Due to the existence of translation gateway which is transparent to the end nodes, new payloads are exchanged during IKE negotiation to empower the end nodes to support E2E-NATPT capability and to detect the presence of NAT-PT gateway. Moreover, we enhanced the Pluto IKEv1 daemon so that it can be able to react once NAT-PT gateway is detected along the communication path. The proposed E2E-NATPT traversal is integrated into StrongSwan IPSec-based solution and is validated by using a Linux-based testbed.

Acknowledgments

The research is funded by Ministry of Science, Technology, and Innovation (MOSTI), Government of Malaysia, under e-ScienceFund grant No. 01-02-01-SF0170.

References

[1] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460 (Proposed Standard) (Dec. 1998).

[2] C. Caicedo and J. Joshi and S. Tuladhar, IPv6 Security Challenges, IEEE Computer 42 (2009) 36–42.

[3] J. Govil, On the Investigation of Transactional and Interoperability Issues between IPv4 and IPv6, in: IEEE International Conference on Electro/Information Technology, 2007, pp. 604 – 609.

[4] S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, RFC 2401 (Proposed Standard), obsoleted by RFC 4301, updated by RFC 3168 (Nov. 1998).

[5] Strongswan 4.5.0, http://www.strongswan.org/ (2010).

[6] S. Kent, R. Atkinson, IP Authentication Header, RFC 2402 (Proposed Standard), obsoleted by RFC 4301, updated by RFC 1826 (Nov. 1998).

[7] S. Kent, R. Atkinson, IP Encapsulating Security Payload (ESP), RFC 2406 (Proposed Standard), obsoleted by RFCs 4303, 4305 (Nov. 1998).

[8] D. Harkins, D. Carrel, The Internet Key Exchange (IKE), RFC 2409 (Proposed Standard), obsoleted by RFC 4306, updated by RFC 4109 (Nov. 1998).

[9] K. Egevang, P. Francis, The IP Network Address Translator (NAT), RFC 1631 (Informational) (May 1994).

[10] M. Smith and R. Hunt, Network Security using NAT and NAPT, in: 10th IEEE International Conference on Networks, 2002, pp. 355 – 360.

[11] B. Aboba, W. Dixon, IPsec-Network Address Translation (NAT) Compatibility Requirements, RFC 3715 (Informational) (Mar. 2004).

[12] S. Kent, K. Seo, Security Architecture for the Internet Protocol, RFC 4301 (Proposed Standard) (Dec. 2005).

[13] G. Montenegro, M. Borella, RSIP Support for End-to-end IPsec, RFC 3104 (Informational) (Oct. 2001).

[14] R. Mahy, P. Matthews, J. Rosenberg, Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN), RFC 5766 (Proposed Standard) (Apr. 2010).

[15] T. Kivinen, B. Swander, A. Huttunen, V. Volpe, Negotiation of NAT-Traversal in the IKE, RFC 3947 (Informational) (Jan. 2005).

[16] A. Huttunen, B. Swander, V. Volpe, L. DiBurro, M. Stenberg, UDP Encapsulation of IPsec ESP Packets, RFC 3948 (Informational) (Jan. 2005).

[17] E. Nordmark, R. Gilligan, Basic Transition Mechanisms for IPv6 Hosts and Routers, RFC 4213 (Proposed Standard) (Oct. 2005).

[18] B. Carpenter, K. Moore, Connection of IPv6 Domains via IPv4 Clouds, RFC 3056 (Informational) (Feb. 2001).

[19] G. Tsirtsis, P. Srisuresh, Network Address Translation - Protocol Translation (NAT-PT), RFC 2766 (Proposed Standard) (Feb. 2000).

[20] Weiping Peng and Yajian Zhou and Cong Wang and Yixian Yang, Research on IPSec-based NAT-PT Transition Mechanism, in: Proceedings of IEEE International Conference on Network Infrastructure and Digital Content, 2009, pp. 222 – 226.

[21] S. Jeong, M.-K. Shin, S. Lee, Applicability Issues of IPSec in NAT-PT,draft-lee-ipsec-nat-pt-applicability-03.txt (Nov. 2008).

[22] S. Jung, J. Choi, Y. Kim, S. Kim, IPSec Support in NAT-PT Scenario for IPv6 Transition, Lecture Notes in Computer Science 3650 (2005) 194–202.

[23] L. Yin and S. Jin and Y. Qifeng and X. Li, T-NATPT - A Novel Proposal for NAT-PT/IPSec Traversing, in: Proceedings of IEEE 2nd International Conference on Anti-counterfeiting, Security and Identification, 2008, pp. 144–148.

[24] D. Piper, The Internet IP Security Domain of Interpretation for ISAKMP, RFC 2407 (Proposed Standard) (Nov. 1998).

[25] Apache HTTP Server Project, http://httpd.apache.org/ (2010).

[26] Bind 9.7.3, http://www.isc.org/software/bind/973 (2010).

[27] Openswan, http://www.openswan.org (2010).

[28] R. Radhakrishnan and M. Jamil and S. Mehfuz and Moinuddin, Security Issues in IPv6, in: Third IEEE International Conference on Networking and Services, 2007, pp. 110–115.

Nazrul M. Ahmad is a Lecturer of Faculty of Information Science & Technology (FIST) at Multimedia University (MMU), Malaysia. He received a M.Sc. in Information Technology from MMU and a B.Eng. in Electronics and Communications Engineering from University of York, UK. His research interests are primarily in communication networks and network security.

Asrul Hadi is a full time lecturer and researcher in computer network and IT security. He is a Certified Linux Professional (LPIC2) and the certified instructor for CEH. Being in the network security field for the past nine years, he has presented at numerous conferences and published articles in several journals. He holds a *Diplôme d'Ingénieur* and *Diplôme d'Étude Approfondie* from ENSEEIHT, *École Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, d'Hydraulique et des Télécommunications*, Toulouse, France. He is currently teaching in Faculty of Information Science and Technology, Multimedia University and pursuing his PhD in network security.