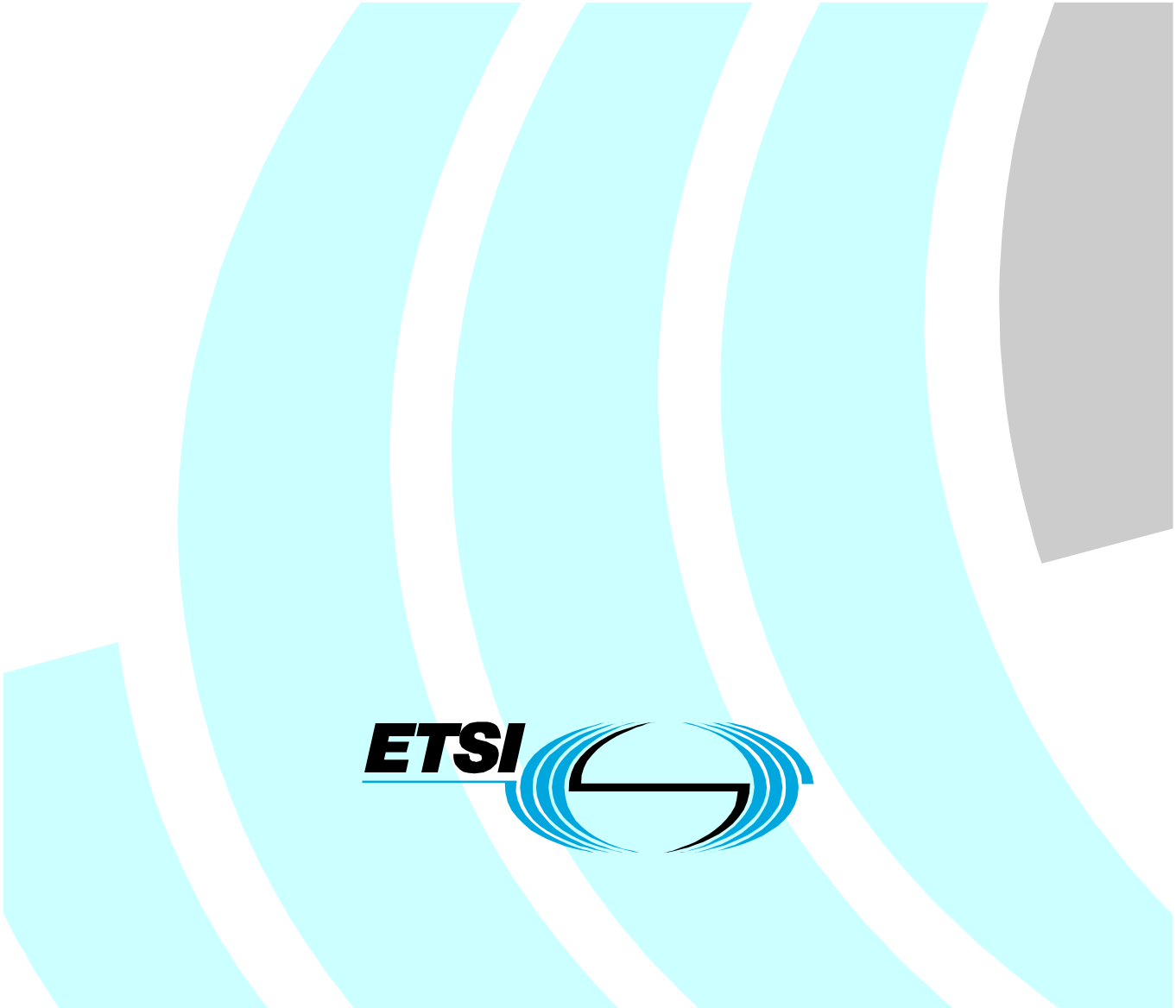
# ETSI TS 102 593 V1.1.1 (2007-04)

*Technical Specification*

**Methods for Testing and Specification (MTS);**
**Internet Protocol Testing (IPT);**
**IPv6 Security;**
**Conformance Test Suite Structure and**
**Test Purposes (TSS&TP)**



Reference
DTS/MTS-IPT-010-IPv6-SecTSS_TP

Keywords
IP, IPv6, security, testing, TSS&TP, TTCN

***ETSI***

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

# Contents

# Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

# Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

# 1 Scope

The purpose of the present document is to provide Test Suite Structure and Test Purposes (TSS&TP) for conformance tests of the security IPv6 protocol based on the requirements defined in the IPv6 requirements catalogue (TS 102 558 [2]) and written according to the guidelines of TS 102 351 [1], ISO/IEC 9646-2 [4] and ETS 300 406 [5].

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

[1] ETSI TS 102 351: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); IPv6 Testing: Methodology and Framework".

[2] ETSI TS 102 558: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT): IPv6 Security; Requirements Catalogue".

[3] ISO/IEC 9646-1: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts".

[4] ISO/IEC 9646-2: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification".

[5] ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

**abstract test case:** Refer to ISO/IEC 9646-1 [3].

**Abstract Test Method (ATM):** Refer to ISO/IEC 9646-1 [3].

**Abstract Test Suite (ATS):** Refer to ISO/IEC 9646-1 [3].

**Implementation Under Test (IUT):** Refer to ISO/IEC 9646-1 [3].

**Lower Tester (LT):** Refer to ISO/IEC 9646-1 [3].

**Test Purpose (TP):** Refer to ISO/IEC 9646-1 [3].

## 3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

| | |
|---|---|
| AH | Authentication Header |
| ATS | Abstract Test Suite |
| ESP | Encapsulating Security Payload |
| IETF | Internet Engineering Task Force |
| IKE | Internet Key Exchange |
| IPv6 | Internet Protocol version 6 |
| IUT | Implementation Under Test |
| RC | Requirements Catalogue |
| RQ | Requirement |
| TP | Test Purpose |
| TSS | Test Suite Structure |
| UDP | User Datagram Protocol |

# 4 Test Suite Structure (TSS)

Test Purposes have been written for IPv6 mobile nodes, correspondent nodes and home agents according to the requirements (RQ) of the requirements catalogue (RC) in TS 102 558 [2]. Test purposes have been written for behaviours requested with "MUST" or "SHOULD", optional behaviour described with "MAY" or similar wording indicating an option has not been turned into test purposes.

The test purposes have been divided into three groups:

Group 1: Authentication Header (AH)

Group 2: Encapsulating Security Payload (ESP)

Group 3: Key Exchange (IKEv2) Protocol

The sub-grouping of these three groups follows the structure of the RC.

Group 1: Authentication Header (AH)

Group 2: Encapsulating Security Payload (ESP)

Group 3: Key Exchange (IKEv2) Protocol

Group 3.1 Exchange Message Structures

Group 3.2 IKE Header and Payload Formats

Group 3.2.1 Configuration payload

Group 3.2.2 IKE Error Types

Group 3.3 IKE Informational Exchanges

Group 3.4 IKE Protocol

Group 3.4.1 Authentication

Group 3.4.1.1 Extensible Authentication Methods

Group 3.4.2 Error Handling

Group 3.4.3 General Protocol Handling

Group 3.4.3.1 Address and Port Agility

Group 3.4.3.2 IP Compression (IPComp)

# Annex A (normative): Test Purposes (TP)

The test purposes have been written in the formal notation TPlan as described in annex A of TS 102 351 [1]. This original textual output ASCII file (SEC.tplan) is contained in archive ts_102593v010101p0.zip which accompanies the present document. The raw text file has been converted to a table format in this annex to allow better readability.

The two formats shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the textual TPlan representation takes precedence over the table format in this annex.

## A.1 Authentication Header (AH)

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2000_01** | | |
| **Summary:** | Test of generating first unicast IPv6 packets with Authentication Header | | |
| **References:** | RQ_002_2000, RQ_002_2006, RQ_002_2011, RQ_002_2013, RQ_002_2015, RQ_002_2017, RQ_002_2027, RQ_002_2032, RQ_002_2033, RQ_002_2034, RQ_002_2036 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2000_01 |

```
with { IUT and destination_node established in an AH_security_association
     }
ensure that
     { when { IUT is requested to send first unicast IPv6Packet
                  containing Authentication_Header }
       then { IUT sends IPv6Packet
                  containing next_header_field of previous_header
                      set to 51
              and containing (Authentication_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                          and containing sequence_number set to 1
                          and containing correctly calculated
                                         Integrity_Check_Value
                                         including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2000_02** | | |
| **Summary:** | Test of generating subsequent unicast IPv6 packets with Authentication Header | | |
| **References:** | RQ_002_2000, RQ_002_2006, RQ_002_2011, RQ_002_2012, RQ_002_2015, RQ_002_2017, RQ_002_2027, RQ_002_2032, RQ_002_2033, RQ_002_2034, RQ_002_2036 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2000_02 |

```
with { IUT and destination_node established in an AH_security_association
     }
ensure that
     { when { IUT is requested to send subsequent unicast IPv6Packet
                  containing Authentication_Header }
       then { IUT sends IPv6Packet
                  containing next_header_field of previous_header
                      set to 51
              and containing (Authentication_Header
                               containing Security_Parameters_Index
                                   set to Security_Parameters_Index
                                          received from destination_node
                                          during SA_establishment
                           and containing sequence_number set to
                              (sequence_number of previous IPv6Packet) plus 1
                           and containing correctly calculated
                                          Integrity_Check_Value
                                          including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2000_03** | | |
| **Summary:** | Test of generating first multicast IPv6 packets with Authentication Header | | |
| **References:** | RQ_002_2000, RQ_002_2007, RQ_002_2011, RQ_002_2013, RQ_002_2015, RQ_002_2017, RQ_002_2027, RQ_002_2032, RQ_002_2033, RQ_002_2034, RQ_002_2036 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2000_03 |

```
with { IUT established in a multicast_group AH_Security_Association
     }
ensure that
     { when { IUT is requested to send first multicast IPv6Packet
                  containing Authentication_Header }
       then { IUT sends IPv6Packet
                  containing next_header_field of previous_header
                      set to 51
              and containing (Authentication_Header
                               containing Security_Parameters_Index
                                          assigned to multicast_group
                                                      Security_Association
                           and containing sequence_number set to 1
                           and containing correctly calculated
                                           Integrity_Check_Value
                                           including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2000_04** | | |
| **Summary:** | Test of generating subsequent multicast IPv6 packets with Authentication Header | | |
| **References:** | RQ_002_2000, RQ_002_2007, RQ_002_2011, RQ_002_2012, RQ_002_2015, RQ_002_2017, RQ_002_2027, RQ_002_2032, RQ_002_2033, RQ_002_2034, RQ_002_2036 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2000_04 |

```
with { IUT established in multicast_group AH_Security_Association
     }
ensure that
     { when { IUT is requested to send subsequent multicast IPv6Packet
                  containing Authentication_Header }
       then { IUT sends IPv6Packet
                  containing next_header_field of previous_header
                      set to 51
              and containing (Authentication_Header
                               containing Security_Parameters_Index
                                   set to Security_Parameters_Index
                                          assigned to multicast_group
                                                      Security_Association
                           and containing sequence_number set to
                              (sequence_number of previous IPv6Packet) plus 1
                           and containing correctly calculated
                                        Integrity_Check_Value
                                        including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2009_01** | | |
| **Summary:** | Test reaction on multicast IPv6 packets for unknown multicast_group SA | | |
| **References:** | RQ_002_2009, RQ_002_2008 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2009_01 |

```
with { IUT established in multicast_group AH_Security_Association
     }
ensure that
     { when { IUT receives multicast IPv6Packet
                  containing (Authentication_Header
                               containing Security_Parameters_Index
                                          unrelated to established
                                          multicast_group Security_Association) }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2042_01** | | |
| **Summary:** | Test reaction on IPv6 packets with AH header and fragmentation header | | |
| **References:** | RQ_002_2042 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2042_01 |

```
with { IUT and destination_node established in an AH_security_association
     }
ensure that
     { when { IUT receives IPv6Packet
                  containing Authentication_Header
              and containing (Fragment_Header
                               containing offset not set to 0) }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2046_01** | | |
| **Summary:** | Test reaction on IPv6 packets with AH header when no SA exists | | |
| **References:** | RQ_002_2046 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2046_01 |

```
with { IUT and destination_node not established in an AH_Security_Association
     }
ensure that
     { when { IUT receives IPv6Packet
                   containing Authentication_Header }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2053_01** | | |
| **Summary:** | Test reaction on IPv6 packets with AH header with incorrect sequence number | | |
| **References:** | RQ_002_2053 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2053_01 |

```
with {     IUT and destination_node established in an AH_security_association
       and IUT and destination_node 'having already exchanged
                                     at least one packet'
     }
ensure that
     { when { IUT receives IPv6Packet
                   containing (Authentication_Header
                               containing sequence_number
                                   set to sequence_number received
                                      in previous IPv6packet) }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2057_01** | | |
| **Summary:** | Test reaction on IPv6 packets with AH header with correct ICV value | | |
| **References:** | RQ_002_2057, RQ_002_2028 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2057_01 |

```
with { IUT and destination_node established in an AH_security_association
     }
ensure that
     { when { IUT receives IPv6Packet
                   containing (Authentication_Header
                               containing Integrity_Check_Value
                               calculated from Security_Association_data
                                          and packet_contents) }
       then { IUT accepts IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_2058_01** | | |
| **Summary:** | Test reaction on IPv6 packets with AH header with incorrect ICV value | | |
| **References:** | RQ_002_2058, RQ_002_2028 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_2058_01 |

```
with { IUT and destination_node established in an AH_security_association
     }
ensure that
     { when { IUT receives IPv6Packet
                  containing (Authentication_Header
                              containing Integrity_Check_Value
                              not calculated from Security_Association_data
                                          and packet_contents) }
       then { IUT discards IPv6Packet }
    }
```

# A.2 Encapsulating Security Payload (ESP)

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3030_01** | | |
| **Summary:** | Test reaction on ESP dummy packet | | |
| **References:** | RQ_002_3030 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3030_01 |

```
with { IUT and destination_node established in an ESP_Security_Association
     }
ensure that
     { when { IUT receives IPv6Packet
                  containing (ESP_Header
                              containing next_header_field set to 59) }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3059_01** | | |
| **Summary:** | Test reaction on IPv6 packets with ESP header when no SA exists | | |
| **References:** | RQ_002_3059 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3059_01 |

```
with { IUT and destination_node established in an ESP_Security_Association
     }
ensure that
     { when { IUT receives IPv6Packet
                  containing ESP_Header
              and containing (Fragment_Header
                              containing offset not set to 0) }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3061_01** | | |
| **Summary:** | Test reaction on IPv6 packets with ESP header when no SA exists | | |
| **References:** | RQ_002_3061, RQ_002_3091 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3061_01 |

```
with { IUT 'has not established ESP Security Association with destination Node'
     }
ensure that
     { when { IUT receives IPv6Packet
                  containing ESP_Header }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3068_01** | | |
| **Summary:** | Test reaction on IPv6 packets with ESP header with correct ICV value | | |
| **References:** | RQ_002_3068, RQ_002_3072 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3068_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and IUT 'having enabled anti-replay service'
     }
ensure that
     { when { IUT receives IPv6Packet
                  containing (ESP_Header
                              containing sequence_number
                              set to sequence_number from received IPv6Packet) }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3077_01** | | |
| **Summary:** | Test reaction on IPv6 packets with ESP header with correct ICV value | | |
| **References:** | RQ_002_3077 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3077_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               combined_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT receives IPv6Packet
                  containing (ESP_Header
                              containing Integrity_Check_Value
                              calculated from Security_Association_data
                                          and packet_contents) }
       then { IUT accepts IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3078_01** | | |
| **Summary:** | Test reaction on IPv6 packets with ESP header with incorrect ICV value | | |
| **References:** | RQ_002_3078, RQ_002_3077 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3078_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               combined_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT receives IPv6Packet
                  containing (ESP_Header
                              containing Integrity_Check_Value
                              not calculated from Security_Association_data
                                          and packet_contents) }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3080_01** | | |
| **Summary:** | Test reaction on IPv6 packets with ESP header with correct ICV value | | |
| **References:** | RQ_002_3080 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3080_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               separate_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT receives IPv6Packet
                 containing (ESP_Header
                             containing Integrity_Check_Value
                             calculated from Security_Association_data
                                          and packet_contents) }
       then { IUT accepts IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3083_01** | | |
| **Summary:** | Test reaction on IPv6 packets with ESP header with incorrect ICV value | | |
| **References:** | RQ_002_3083, RQ_002_3080 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3083_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               separate_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT receives IPv6Packet
                 containing (ESP_Header
                             containing Integrity_Check_Value
                             not calculated from Security_Association_data
                                          and packet_contents) }
       then { IUT discards IPv6Packet }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3102_01** | | |
| **Summary:** | Test of generating first unicast IPv6 packets with ESP Header, transport mode | | |
| **References:** | RQ_002_3102, RQ_002_3004, RQ_002_3005, RQ_002_3009, RQ_002_3012, RQ_002_3027, RQ_002_3037, RQ_002_3113 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3102_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               separate_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT is requested to send first IPv6Packet in transport_mode
                  containing ESP_Header }
       then { IUT sends IPv6Packet in transport_mode
                  containing next_header_field of previous_header
                      set to 50
              and containing (ESP_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                              and containing sequence_number set to 1
                              and containing necessary padding_bytes
                              and containing pad_length
                                  set to number of padding_bytes
                              and containing correctly calculated
                                             Integrity_Check_Value
                                  including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3102_02** | | |
| **Summary:** | Test of generating subsequent unicast IPv6 packets with ESP Header, transport mode | | |
| **References:** | RQ_002_3102, RQ_002_3004, RQ_002_3005, RQ_002_3006, RQ_002_3009, RQ_002_3027, RQ_002_3037, RQ_002_3112 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3102_02 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               separate_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT is requested to send subsequent IPv6Packet in transport_mode
                  containing ESP_Header }
       then { IUT sends IPv6Packet in transport_mode
                  containing next_header_field of previous_header
                      set to 50
              and containing (ESP_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                              and containing sequence_number set to
                                  (sequence_number of previous IPv6Packet) plus 1
                              and containing necessary padding_bytes
                              and containing pad_length
                                  set to number of padding_bytes
                              and containing correctly calculated
                                             Integrity_Check_Value
                                  including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3103_01** | | |
| **Summary:** | Test of generating first unicast IPv6 packets with ESP Header, tunnel mode | | |
| **References:** | RQ_002_3103, RQ_002_3004, RQ_002_3005, RQ_002_3009, RQ_002_3012, RQ_002_3027, RQ_002_3037, RQ_002_3092, RQ_002_3113 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3103_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               separate_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT is requested to send first IPv6Packet in tunnel_mode
                  containing ESP_Header }
       then { IUT sends IPv6Packet in tunnel_mode
                  containing next_header_field of previous_header
                      set to 50
              and containing (ESP_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                          and containing sequence_number set to 1
                          and containing necessary padding_bytes
                          and containing pad_length
                                  set to number of padding_bytes
                          and containing correctly calculated
                                         Integrity_Check_Value
                              including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3103_02** | | |
| **Summary:** | Test of generating subsequent unicast IPv6 packets with ESP Header, tunnel mode | | |
| **References:** | RQ_002_3103, RQ_002_3004, RQ_002_3005, RQ_002_3006, RQ_002_3009, RQ_002_3027, RQ_002_3037, RQ_002_3092, RQ_002_3112 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3103_02 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               separate_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT is requested to send subsequent IPv6Packet in tunnel_mode
                  containing ESP_Header }
       then { IUT sends IPv6Packet in tunnel_mode
                  containing next_header_field of previous_header
                      set to 50
              and containing (ESP_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                          and containing sequence_number set to
                              (sequence_number of previous IPv6Packet) plus 1
                          and containing necessary padding_bytes
                          and containing pad_length
                                  set to number of padding_bytes
                          and containing correctly calculated
                                         Integrity_Check_Value
                              including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3107_01** | | |
| **Summary:** | Test of generating first unicast IPv6 packets with ESP Header, transport mode | | |
| **References:** | RQ_002_3102, RQ_002_3004, RQ_002_3005, RQ_002_3009, RQ_002_3012, RQ_002_3027, RQ_002_3113 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3107_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               combined_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT is requested to send first IPv6Packet in transport_mode
                  containing ESP_Header }
       then { IUT sends IPv6Packet in transport_mode
                  containing next_header_field of previous_header
                      set to 50
              and containing (ESP_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                              and containing sequence_number set to 1
                              and containing necessary padding_bytes
                              and containing pad_length
                                  set to number of padding_bytes
                              and containing correctly calculated
                                             Integrity_Check_Value
                                  including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3107_02** | | |
| **Summary:** | Test of generating subsequent unicast IPv6 packets with ESP Header, transport mode | | |
| **References:** | RQ_002_3107, RQ_002_3004, RQ_002_3005, RQ_002_3006, RQ_002_3009, RQ_002_3027, RQ_002_3112 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3107_02 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               combined_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT is requested to send subsequent IPv6Packet in transport_mode
                  containing ESP_Header }
       then { IUT sends IPv6Packet in transport_mode
                  containing next_header_field of previous_header
                      set to 50
              and containing (ESP_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                              and containing sequence_number set to
                                  (sequence_number of previous IPv6Packet) plus 1
                              and containing necessary padding_bytes
                              and containing pad_length
                                  set to number of padding_bytes
                              and containing correctly calculated
                                             Integrity_Check_Value
                                  including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3108_01** | | |
| **Summary:** | Test of generating first unicast IPv6 packets with ESP Header, tunnel mode | | |
| **References:** | RQ_002_3108, RQ_002_3004, RQ_002_3005, RQ_002_3009, RQ_002_3012, RQ_002_3027, RQ_002_3092, RQ_002_3113 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3108_01 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               combined_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT is requested to send first IPv6Packet in tunnel_mode
                  containing ESP_Header }
       then { IUT sends IPv6Packet in tunnel_mode
                  containing next_header_field of previous_header
                      set to 50
              and containing (ESP_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                              and containing sequence_number set to 1
                              and containing necessary padding_bytes
                              and containing pad_length
                                  set to number of padding_bytes
                              and containing correctly calculated
                                             Integrity_Check_Value
                                  including necessary padding_bits) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_3108_02** | | |
| **Summary:** | Test of generating subsequent unicast IPv6 packets with ESP Header, tunnel mode | | |
| **References:** | RQ_002_3108, RQ_002_3004, RQ_002_3005, RQ_002_3006, RQ_002_3009, RQ_002_3027, RQ_002_3092, RQ_002_3112 | | |
| **IUT Role** | Ipsec_host | **Test Case:** | TC_SEC_3108_02 |

```
with {     IUT and destination_node established in an ESP_Security_Association
       and ESP_Security_Association configured to use
               combined_confidentiality_and_integrity_algorithms
     }
ensure that
     { when { IUT is requested to send subsequent IPv6Packet in tunnel_mode
                  containing ESP_Header }
       then { IUT sends IPv6Packet in tunnel_mode
                  containing next_header_field of previous_header
                      set to 50
              and containing (ESP_Header
                              containing Security_Parameters_Index
                                  set to Security_Parameters_Index
                                         received from destination_node
                                         during SA_establishment
                              and containing sequence_number set to
                                  (sequence_number of previous IPv6Packet) plus 1
                              and containing necessary padding_bytes
                              and containing pad_length
                                  set to number of padding_bytes
                              and containing correctly calculated
                                             Integrity_Check_Value
                                  including necessary padding_bits) }
    }
```

# A.3 Key Exchange (IKEv2) Protocol

## A.3.1 Exchange Message Structures

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6400_01** | | |
| **Summary:** | Test of generating IKE_SA_INIT request | | |
| **References:** | RQ_002_6400, RQ_002_6034, RQ_002_6077, RQ_002_6084, RQ_002_6085, RQ_002_6086, RQ_002_6128, RQ_002_6129, RQ_002_6232, RQ_002_6236, RQ_002_6240, RQ_002_6250, RQ_002_6263, RQ_002_6304, RQ_002_6344 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6400_01 |

```
with { IUT ready to establish a Security_Association using IKEv2
     }
ensure that
     { when { IUT is requested to send IKE_SA_INIT_request }
       then { IUT sends IKE_SA_INIT_request
                  containing (IKE_Header
                              containing IKE_SA_Initiators_SPI not set to 0
                            and containing IKE_SA_Responders_SPI set to 0
                            and containing Major_Version set to 2
                            and containing Exchange_Type set to 34 IKE_SA_INIT
                            and containing Flags set to 00010000'B'
                            and containing Message_ID set to 0)
              and containing (Security_Association_payload
                              containing at least 1 Proposal
                                         containing at least 1 Transform)
              and containing Key_Exchange_payload
              and containing (Nonce_payload
                              containing Nonce_Data
                                      of at least 128 bits
                                     and 'at least half the prf key length') }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6401_01** | | |
| **Summary:** | Test reaction on IKE_SA_INIT request | | |
| **References:** | RQ_002_6401, RQ_002_6036, RQ_002_6232, RQ_002_6233, RQ_002_6236, RQ_002_6240, RQ_002_6250, RQ_002_6263, RQ_002_6304, RQ_002_6344 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6401_01 |

```
with { IUT ready to establish Security_Association using IKEv2
     }
ensure that
     { when { IUT receives IKE_SA_INIT_request }
       then { IUT sends IKE_SA_INIT_response
                  containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                         received in IKE_SA_INIT_request
                             and containing IKE_SA_Responders_SPI not set to 0
                             and containing Major_Version set to 2
                             and containing Exchange_Type set to 34 IKE_SA_INIT
                             and containing Flags set to 00000100'B'
                             and containing Message_ID
                                  set to Message_ID
                                         received in IKE_SA_INIT_request)
              and containing (Security_Association_payload
                              containing 1 proposal
                                         received in IKE_SA_INIT_request)
              and containing Key_Exchange_payload
              and containing Nonce_payload }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6403_01** | | |
| **Summary:** | Test of generating IKE_AUTH request | | |
| **References:** | RQ_002_6403, RQ_002_6034, RQ_002_6084, RQ_002_6085, RQ_002_6086, RQ_002_6232, RQ_002_6233, RQ_002_6236, RQ_002_6240, RQ_002_6250, RQ_002_6263, RQ_002_6310, RQ_002_6430, RQ_002_6431 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6403_01 |

```
with {     IUT having sent IKE_SA_INIT_request
       and IUT having received IKE_SA_INIT_response
     }
ensure that
     { when { IUT is requested to send IKE_AUTH_request }
       then { IUT sends IKE_AUTH_request
                  containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                         received in IKE_SA_INIT_request
                              and containing IKE_SA_Responders_SPI
                                  set to IKE_SA_Responders_SPI
                                         received in IKE_SA_INIT_response
                              and containing Major_Version set to 2
                              and containing Exchange_Type set to 35 IKE_AUTH
                              and containing Flags set to 00010000'B'
                              and containing Message_ID set to 1)
              and containing (Encrypted_payload
                              containing Identification_payload_initiator
                                        'Next Payload field of previous
                                         payload is set to 35'
                              and containing Authentication_payload
                              and containing(Security_Association_payload
                                         containing at least 1 proposal
                                              containing at least 1 transform)
                              and containing Traffic_Selector_payload_initiator
                                        'Next Payload field of previous
                                         payload is set to 44'
                              and containing Traffic_Selector_payload_responder
                                        'Next Payload field of previous
                                         payload is set to 45') }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6405_01** | | |
| **Summary:** | Test reaction on IKE_AUTH request | | |
| **References:** | RQ_002_6405, RQ_002_6036, RQ_002_6232, RQ_002_6233, RQ_002_6236, RQ_002_6240, RQ_002_6250, RQ_002_6263, RQ_002_6312, RQ_002_6430, RQ_002_6431 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6405_01 |

```
with {    IUT having received IKE_SA_INIT_request
      and IUT having sent IKE_SA_INIT_response
     }
ensure that
     { when { IUT receives IKE_AUTH_request }
       then { IUT sends IKE_AUTH_response
                  containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                         received in IKE_SA_INIT_request
                              and containing IKE_SA_Responders_SPI
                                  set to IKE_SA_Responders_SPI
                                         sent in IKE_SA_INIT_response
                              and containing Major_Version set to 2
                              and containing Exchange_Type set to 35 IKE_AUTH
                              and containing Flags set to 00000100'B'
                              and containing Message_ID
                                  set to Message_ID
                                         received in IKE_AUTH_request)
              and containing (Encrypted_payload
                              containing Identification_payload_responder
                                        'Next Payload field of previous payload
                                         is set to 36'
                              and containing Authentication_payload
                              and containing (Security_Association_payload
                                               containing 1 proposal
                                                      received in IKE_AUTH_request)
                              and containing Traffic_Selector_payload_initiator
                                        'Next Payload field of previous payload
                                         is set to 44'
                              and containing Traffic_Selector_payload_responder
                                        'Next Payload field of previous payload
                                         is set to 45' }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6407_01** | | |
| **Summary:** | Test of generating CREATE_CHILD_SA request | | |
| **References:** | RQ_002_6407, RQ_002_6035, RQ_002_6084, RQ_002_6085, RQ_002_6086, RQ_002_6128, RQ_002_6129, RQ_002_6232, RQ_002_6233, RQ_002_6236, RQ_002_6240, RQ_002_6250, RQ_002_6263, RQ_002_6344 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6407_01 |

```
with {     IUT having completed IKE_SA_INIT exchange
       and IUT having completed IKE_AUTH exchange
     }
ensure that
     { when { IUT is requested to send CREATE_CHILD_SA_request }
       then { IUT sends CREATE_CHILD_SA_request
                  containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                    sent or received in the IKE_SA_INIT_request
                             and containing IKE_SA_Responders_SPI
                                  set to IKE_SA_Responders_SPI
                                    sent or received in the IKE_SA_INIT_response
                             and containing Major_Version set to 2
                             and containing Exchange_Type set to 36 CREATE_CHILD_SA
                             and containing Flags set to 00010000'B'
                             and containing Message_ID
                                  set to previous sent Message_ID plus 1)
              and containing (Encrypted_payload
                              containing (Security_Association_payload
                                           containing at least 1 proposal
                                                 containing at least 1 transform)
                             and containing (Nonce_payload
                                           containing Nonce_Data
                                                 of at least 128 bits
                                              and 'at least half the
                                                   prf key length')
                             and containing Traffic_Selector_payload_initiator
                                           'Next Payload field of previous
                                            payload is set to 44'
                             and containing Traffic_Selector_payload_responder
                                           'Next Payload field of previous
                                            payload is set to 45')}
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6409_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request | | |
| **References:** | RQ_002_6409, RQ_002_6036, RQ_002_6232, RQ_002_6233, RQ_002_6236, RQ_002_6240, RQ_002_6250, RQ_002_6263, RQ_002_6344 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6409_01 |

```
with {     IUT having completed IKE_SA_INIT exchange
       and IUT having completed IKE_AUTH exchange
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request }
       then { IUT sends CREATE_CHILD_SA_response
                  containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                  sent or received in the IKE_SA_INIT_request
                             and containing IKE_SA_Responders_SPI
                                  set to IKE_SA_Responders_SPI
                                  sent or received in the IKE_SA_INIT_request
                             and containing Major_Version set to 2
                             and containing Exchange_Type set to 36 CREATE_CHILD_SA
                             and containing Flags set to 00000100'B'
                             and containing Message_ID
                                  set to Message_ID
                                         received in CREATE_CHILD_SA_request)
              and containing (Encrypted_payload
                              containing (Security_Association_payload
                                           containing 1 proposal
                                             received in CREATE_CHILD_SA_request)
                             and containing Nonce_payload
                             and containing Traffic_Selector_payload_initiator
                                           'Next Payload field of previous
                                            payload is set to 44'
                             and containing Traffic_Selector_payload_responder
                                           'Next Payload field of previous
                                            payload is set to 45')}
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6411_01** | | |
| **Summary:** | Test of generating INFORMATIONAL_request | | |
| **References:** | RQ_002_6411, RQ_002_6035, RQ_002_6232, RQ_002_6233, RQ_002_6236, RQ_002_6240, RQ_002_6250 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6411_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT is requested to send INFORMATIONAL_request }
       then { IUT sends INFORMATIONAL_request
                  containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                  sent or received in the IKE_SA_INIT_request
                             and containing IKE_SA_Responders_SPI
                                  set to IKE_SA_Responders_SPI
                                  sent or received in the IKE_SA_INIT_request
                             and containing Major_Version set to 2
                             and containing Exchange_Type set to 37 INFORMATIONAL
                             and containing Flags set to 00010000'B'
                             and containing Message_ID
                                  set to previous sent Message_ID plus 1)
              and containing (Encrypted_payload
                              containing 0 or more Notify_payload
                             and containing 0 or more Delete_payload
                             and containing 0 or more Configuration_payload) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6412_01** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request | | |
| **References:** | RQ_002_6412, RQ_002_6036, RQ_002_6232, RQ_002_6233, RQ_002_6236, RQ_002_6240, RQ_002_6250 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6412_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_request }
       then { IUT sends INFORMATIONAL_response
                  containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                  sent or received in the IKE_SA_INIT_request
                             and containing IKE_SA_Responders_SPI
                                  set to IKE_SA_Responders_SPI
                                  sent or received in the IKE_SA_INIT_request
                             and containing Major_Version set to 2
                             and containing Exchange_Type set to 37 INFORMATIONAL
                             and containing Flags set to 00000100'B'
                             and containing Message_ID
                                  set to Message_ID
                                         received in INFORMATIONAL_request)
              and containing (Encrypted_payload
                              containing 0 or more Notify_payload
                             and containing 0 or more Delete_payload
                             and containing 0 or more Configuration_payload) }
    }
```

## A.3.2 IKE Header and Payload Formats

### A.3.2.1 Configuration payload

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6468_01** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request with unsupported Configuration payload | | |
| **References:** | RQ_002_6468 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6468_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when {    IUT receives INFORMATIONAL_request
                     containing (Configuration_payload
                                 containing Configuration_Type
                                     set to 1 CFG_REQUEST
                                and containing any unsupported
                                               Configuration_Attribute) }
       then {    IUT sends INFORMATIONAL_response
                     containing (Configuration_payload
                                 containing Configuration_Type
                                     set to 2 CFG_REPLY
                                and not containing any unsupported
                                               Configuration_Attribute)
              or not containing (Configuration_payload) }
    }
```

### A.3.2.2 IKE Error Types

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6365_01** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request containing incorrect value | | |
| **References:** | RQ_002_6365, RQ_002_6368 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6365_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_request
                  containing 'syntactically incorrect value' }
       then { IUT sends INFORMATIONAL_response
                  containing (Encrypted_payload
                              containing Notify_payload
                                         containing Notify_Message_Type
                                             set to 7 INVALID_SYNTAX) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6375_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request containing Traffic Selectors indicating address range | | |
| **References:** | RQ_002_6375 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6375_01 |

```
with {     IUT having established an IKE_Security_Association
       and IUT 'only supporting Traffic Selectors specifying a
                single pair of addresses'
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                containing (Traffic_Selector_payload
                            containing Traffic_Selector
                                indicating 'address range') }
       then { IUT sends CREATE_CHILD_SA_response
                containing (Notify_payload
                            containing Notify_Message_Type
                                set to 34 SINGLE_PAIR_REQUIRED) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6376_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request when no more CHILD_SA can be established | | |
| **References:** | RQ_002_6376 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6376_01 |

```
with {     IUT having established an IKE_Security_Association
       and IUT 'unable to establish any further CHILD_SA'
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request }
       then { IUT sends CREATE_CHILD_SA_response
                containing (Notify_payload
                            containing Notify_Message_Type
                                set to 35 NO_ADDITIONAL_SAS) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6379_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request containing unacceptable Traffic Selectors | | |
| **References:** | RQ_002_6379 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6379_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                containing (Traffic_Selector_payload
                            containing 1 or more
                                       unacceptable Traffic_Selector) }
       then { IUT sends CREATE_CHILD_SA_response
                containing (Notify_payload
                            containing Notify_Message_Type
                                set to 38 TS_UNACCEPTABLE) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6393_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request containing transport mode request | | |
| **References:** | RQ_002_6393 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6393_01 |

```
with {     IUT having established an IKE_Security_Association
       and IUT 'ready to accept transport mode request'
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                containing (Notify_payload
                            containing Notify_Message_Type
                                set to 16391 USE_TRANSPORT_MODE) }
       then { IUT sends CREATE_CHILD_SA_response
                containing (Notify_payload
                            containing Notify_Message_Type
                                set to 16391 USE_TRANSPORT_MODE) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6394_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request containing transport mode request | | |
| **References:** | RQ_002_6394 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6394_01 |

```
with {     IUT having established an IKE_Security_Association
       and IUT 'not ready to accept transport mode request'
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                containing (Notify_payload
                            containing Notify_Message_Type
                                set to 16391 USE_TRANSPORT_MODE) }
       then { IUT sends CREATE_CHILD_SA_response
              not containing (Notify_payload
                            containing Notify_Message_Type
                                set to 16391 USE_TRANSPORT_MODE) }
    }
```

## A.3.3 IKE Informational Exchanges

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6007_01** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request without payload | | |
| **References:** | RQ_002_6007, RQ_002_6012 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6007_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_request
                not containing a payload }
       then { IUT sends INFORMATIONAL_response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6014_01** | | |
| **Summary:** | Test of generating INFORMATIONAL_request with Delete payload for IKE_SA | | |
| **References:** | RQ_002_6014, RQ_002_6016, RQ_002_6062, RQ_002_6064, RQ_002_6415,RQ_002_6416, RQ_002_6417 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6014_01 |

```
with {     IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT is requested to send INFORMATIONAL_request
                  containing Delete_payload }
       then { IUT sends INFORMATIONAL_request
                  containing IKE_Header
              and containing (Encrypted_payload
                              containing Delete_payload
                                         containing Protocol_ID indicating 1
                                     and containing SPI_Size indicating 0
                                     and not containing SPI) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6014_02** | | |
| **Summary:** | Test of generating INFORMATIONAL_request with Delete payload for CHILD_SA | | |
| **References:** | RQ_002_6014, RQ_002_6016, RQ_002_6060, RQ_002_6061, RQ_002_6415,RQ_002_6416, RQ_002_6417 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6014_02 |

```
with {     IUT having established an IKE_Security_Association
       and IUT having established at least 1 CHILD_SA
     }
ensure that
     { when { IUT is requested to send INFORMATIONAL_request
                  containing Delete_payload }
       then { IUT sends INFORMATIONAL_request
                  containing IKE_Header
              and containing (Encrypted_payload
                              containing Delete_payload
                                         containing Protocol_ID indicating 2 or
3
                                     and containing SPI_Size indicating 4
                                 and containing SPI) }     }
```

# A.3.4 IKE Protocol

## A.3.4.1 Authentication

### A.3.4.1.1 Extensible Authentication Methods

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6151_01** | | |
| **Summary:** | Test of generating IKE_AUTH request for extensible authentication methods, message 3 | | |
| **References:** | RQ_002_6151 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6151_01 |

```
with { ordered (     IUT having sent IKE_SA_INIT_request
                   and IUT having received IKE_SA_INIT_response )
       and IUT configured 'to use extensible authentication methods'
     }
ensure that
     { when { IUT is requested to send IKE_AUTH_request }
       then { IUT sends IKE_AUTH_request
              not containing Authentication_payload }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6152_01** | | |
| **Summary:** | Test reaction on IKE_AUTH request for extensible authentication methods, message 3 | | |
| **References:** | RQ_002_6152, RQ_002_6153 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6152_01 |

```
with { ordered (     IUT having received IKE_SA_INIT_request
                   and IUT having sent IKE_SA_INIT_response )
       and IUT configured 'to support extensible authentication methods'
     }
ensure that
     { when {     IUT receives IKE_AUTH_request
                  not containing Authentication_payload }
       then {     IUT sends IKE_AUTH_response
                      containing Extensible_Authentication_Protocol_payload
                  and containing Identification_payload
                  and containing Authentication_payload
              and not containing Security_Association_payload
              and not containing any Traffic_Selector_payload }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6153_01** | | |
| **Summary:** | Test of generating IKE_AUTH request for extensible authentication methods, message 5 | | |
| **References:** | RQ_002_6153 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6153_01 |

```
with { ordered (     IUT having sent IKE_SA_INIT_request         'message 1'
                   and IUT having received IKE_SA_INIT_response  'message 2'
                   and IUT having sent IKE_AUTH_request          'message 3'
                   and IUT having received IKE_AUTH_response     'message 4' )
       and IUT configured 'to use extensible authentication'
     }
ensure that
     { when { IUT is requested to send IKE_AUTH_request }
       then { IUT sends IKE_AUTH_request
                  containing Extensible_Authentication_Protocol_payload }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6161_01** | | |
| **Summary:** | Test reaction on IKE_AUTH request for extensible authentication methods, message 5 | | |
| **References:** | RQ_002_6161 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6161_01 |

```
with { ordered (     IUT having received IKE_SA_INIT_request  'message 1'
                 and IUT having sent IKE_SA_INIT_response     'message 2'
                 and IUT having received IKE_AUTH_request     'message 3'
                 and IUT having sent IKE_AUTH_response        'message 4' )
       and IUT having completed 'authentication method successfully'
     }
ensure that
     { when { IUT receives IKE_AUTH_request
                 containing Extensible_Authentication_Protocol_payload }
       then { IUT sends IKE_AUTH_response
                 containing (Extensible_Authentication_Protocol_payload
                             containing Code set to 3 'success' }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6162_01** | | |
| **Summary:** | Test reaction on IKE_AUTH request for extensible authentication methods, message 5 | | |
| **References:** | RQ_002_6162, RQ_002_6374 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6162_01 |

```
with { ordered (     IUT having received IKE_SA_INIT_request    'message 1'
                 and IUT having sent IKE_SA_INIT_response       'message 2'
                 and IUT having received IKE_AUTH_request       'message 3'
                 and IUT sent IKE_AUTH_response                 'message 4' )
       and IUT having completed 'authentication method unsuccessfully'
     }
ensure that
     { when { IUT receives IKE_AUTH_request
                 containing Extensible_Authentication_Protocol_payload }
       then { IUT sends IKE_AUTH_response
                 containing (Notify_payload
                             containing Notify_Message_Type
                                 set to 24 AUTHENTICATION_FAILED) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6164_01** | | |
| **Summary:** | Test of generating IKE_AUTH request for extensible authentication methods, message 7 | | |
| **References:** | RQ_002_6164 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6164_01 |

```
with { ordered (     IUT having sent IKE_SA_INIT_request         'message 1'
                 and IUT having received IKE_SA_INIT_response    'message 2'
                 and IUT having sent IKE_AUTH_request            'message 3'
                 and IUT having received IKE_AUTH response       'message 4'
                 and IUT having sent IKE_AUTH_request            'message 5'
                 and IUT having received IKE_AUTH_response       'message 6' )
       and IUT 'ready to finalize extensible authentication'
     }
ensure that
     { when { IUT is requested to send IKE_AUTH_request }
       then { IUT sends IKE_AUTH_request
                 containing Authentication_payload }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6164_02** | | |
| **Summary:** | Test reaction on IKE_AUTH request for extensible authentication methods, message 7 | | |
| **References:** | RQ_002_6164 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6164_02 |

```
with { ordered (     IUT having received IKE_SA_INIT_request    'message 1'
                   and IUT having sent IKE_SA_INIT_response      'message 2'
                   and IUT having received IKE_AUTH_request      'message 3'
                   and IUT having sent IKE_AUTH_response         'message 4'
                   and IUT having received IKE_AUTH_request      'message 5'
                   and IUT having sent IKE_AUTH_response         'message 6' )
       and IUT having completed 'authentication method successfully'
     }
ensure that
     { when { IUT receives IKE_AUTH_request
                   containing Authentication_payload }
       then { IUT sends IKE_AUTH_response
                   containing Authentication_payload
              and containing Security_Association_payload
              and containing Traffic_Selector_payload_initiator
                                 'Next Payload field of previous
                                  payload has value 44'
              and containing Traffic_Selector_payload_responder
                                 'Next Payload field of previous
                                   payload has value 45' }
    }
```

## A.3.4.2 Error Handling

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6186_01** | | |
| **Summary:** | Test reaction on badly formatted IKE_SA_INIT request | | |
| **References:** | RQ_002_6186 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6186_01 |

```
with {     IUT ready to receive IKE_SA_INIT_request
       and IUT ready to send IKE_SA_INIT response
     }
ensure that
     { when { IUT receives badly formatted IKE_SA_INIT_request }
       then { IUT sends IKE_SA_INIT_response
                   containing Notify_payload }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6186_02** | | |
| **Summary:** | Test reaction on badly formatted IKE_AUTH request | | |
| **References:** | RQ_002_6186 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6186_02 |

```
with { ordered (     IUT having received IKE_SA_INIT_request
                   and IUT having sent IKE_SA_INIT_response
     }
ensure that
     { when { IUT receives badly formatted IKE_AUTH_request }
       then { IUT sends IKE_AUTH_response
                   containing Notify_payload }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6188_01** | | |
| **Summary:** | Test reaction on badly formatted IKE_SA_INIT response | | |
| **References:** | RQ_002_6188 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6188_01 |

```
with { IUT having sent IKE_SA_INIT_request
     }
ensure that
     { when { IUT receives badly formatted IKE_SA_INIT_response }
       then { IUT sends no response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6188_02** | | |
| **Summary:** | Test reaction on badly formatted IKE_AUTH response | | |
| **References:** | RQ_002_6188 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6188_02 |

```
with { ordered (     IUT having sent IKE_SA_INIT_request
                 and IUT having received IKE_SA_INIT_response
                 and IUT having sent IKE_AUTH_request )
     }
ensure that
     { when { IUT receives badly formatted IKE_AUTH_response }
       then { IUT sends no response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6189_01** | | |
| **Summary:** | Test reaction on request outside of known IKE_SA | | |
| **References:** | RQ_002_6189, RQ_002_6190, RQ_002_6191 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6189_01 |

```
with { IUT having no IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request on UDP_port_500 }
       then { IUT sends CREATE_CHILD_SA_response on UDP_port_500
                   containing destination_address
                       set to source_address
                              received in CREATE_CHILD_SA_request
              and containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                         received in CREATE_CHILD_SA_request
                              and containing IKE_SA_Responders_SPI
                                  set to IKE_SA_Responders_SPI
                                         received in CREATE_CHILD_SA_request
                              and containing Message_ID
                                  set to Message_ID
                                         received in CREATE_CHILD_SA_request)
              and not containing an Encrypted_payload
              and containing (Notify_payload      -- Not encrypted
                              containing Notify_Message_Type
                                  set to 4 INVALID_IKE_SPI ) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6189_02** | | |
| **Summary:** | Test reaction on request outside of known IKE_SA | | |
| **References:** | RQ_002_6189, RQ_002_6190, RQ_002_6191 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6189_02 |

```
with { IUT having no IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_request on UDP_port_4500 }
       then { IUT sends INFORMATIONAL_response on UDP_port_4500
                   containing destination_address
                       set to source_address received in INFORMATIONAL_request
              and containing (IKE_Header
                              containing IKE_SA_Initiators_SPI
                                  set to IKE_SA_Initiators_SPI
                                         received in INFORMATIONAL_request
                              and containing IKE_SA_Responders_SPI
                                  set to IKE_SA_Responders_SPI
                                          received in INFORMATIONAL_request
                              and containing Message_ID
                                  set to Message_ID
                                         received in INFORMATIONAL_request
              and not containing an Encrypted_payload
              and containing (Notify_payload      -- Not encrypted
                              containing Notify_Message_Type
                                  set to 4 INVALID_IKE_SPI) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6023_01** | | |
| **Summary:** | Test reaction on cryptographically unprotected response indicating invalid SPI | | |
| **References:** | RQ_002_6023, RQ_002_6194 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6023_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_response
                   containing (IKE_Header
                              containing unknown IKE_SA_Initiators_SPI
                             and containing unknown IKE_SA_Responders_SPI)
              and not containing an Encrypted_payload
              and containing (Notify_payload      -- Not encrypted
                              containing Notify_Message_Type
                                  set to 4 INVALID_IKE_SPI) }
       then { IUT sends no response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6023_02** | | |
| **Summary:** | Test reaction on cryptographically unprotected response indicating invalid SPI | | |
| **References:** | RQ_002_6023, RQ_002_6194 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6023_02 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_response
                  containing (IKE_Header
                              containing unknown IKE_SA_Initiators_SPI
                          and containing unknown IKE_SA_Responders_SPI)
              and not containing an Encrypted_payload
              and containing (Notify_payload      -- Not encrypted
                              containing Notify_Message_Type
                                  set to 4 INVALID_IKE_SPI) }
       then { IUT sends no response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6023_03** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request with Notify payload without cryptographic protection | | |
| **References:** | RQ_002_6023, RQ_002_6022 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6023_03 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_request
              not containing an Encrypted_payload
                  containing (Notify_payload      -- Not encrypted
                              containing Notify_Message_Type
                                  set to 4 INVALID_IKE_SPI) }
       then { IUT sends no INFORMATIONAL_response }
    }
```

## A.3.4.3 General Protocol Handling

### A.3.4.3.1 Address and Port Agility

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6206_01** | | |
| **Summary:** | Test reaction on IKE_SA_INIT request received on UDP port other than 500 or 4500 | | |
| **References:** | RQ_002_6206, RQ_002_6131, RQ_002_6212 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6206_01 |

```
with {     IUT ready to receive IKE_SA_INIT_request
       and IUT ready to send IKE_SA_INIT_response
     }
ensure that
     { when { IUT receives IKE_SA_INIT_request not on UDP_port_500
                                          and not on UDP_port_4500 }
       then { IUT sends IKE_SA_INIT_response on 'UDP port on which request
                                                 was received' }
    }
```

### A.3.4.3.2 IP Compression (IPComp)

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6385_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with compression offer | | |
| **References:** | RQ_002_6385, RQ_002_6203 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6385_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                 containing IKE_Header
            and containing (Notify_payload
                            containing Notify_Message_Type
                                set to 16387 IPCOMP_SUPPORTED
                       and containing (Notification_Data
                                       containing transform_ID)
            and containing additional (Notify_payload
                            containing Notify_Message_Type
                                set to 16387 IPCOMP_SUPPORTED
                       and containing (Notification_Data
                                       containing transform_ID) }
       then { IUT sends CREATE_CHILD_SA_response
                 containing IKE_Header
            and optionally (containing (Notify_payload
                                       containing Notify_Message_Type
                                           set to 16387 IPCOMP_SUPPORTED
                                  and containing (Notification_Data
                                                  containing 1 transform_ID
                                                    received in
                                                    CREATE_CHILD_SA_request)
            and not containing additional (Notify_payload
                                           containing Notify_Message_Type
                                               set to 16387 IPCOMP_SUPPORTED)
}
    }
```

### A.3.4.3.3 Message Format

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6369_01** | | |
| **Summary:** | Test reaction on request with incorrect Message ID | | |
| **References:** | RQ_002_6369, RQ_002_6370 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6369_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when {     IUT receives CREATE_CHILD_SA_request
                      containing (IKE_Header
                                  containing Message_ID 'out of sequence') }
       then {     IUT not sends CREATE_CHILD_SA_response
                  and IUT optionally sends INFORMATIONAL_request
                      containing (Notify_payload
                                  containing Notify_Message_Type
                                      set to 9 INVALID_MESSAGE_ID) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6369_02** | | |
| **Summary:** | Test reaction on request with incorrect Message ID | | |
| **References:** | RQ_002_6369, RQ_002_6370 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6369_02 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when {     IUT receives INFORMATIONAL_request
                      containing (IKE_Header
                                      containing Message_ID 'out of sequence' }
       then {     IUT not sends INFORMATIONAL_response
                  and IUT optionally sends INFORMATIONAL_request
                            containing (Notify_payload
                                            containing Notify_Message_Type
                                                set to 9 INVALID_MESSAGE_ID) }
    }
```

### A.3.4.3.4 Overlapping Requests

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6041_01** | | |
| **Summary:** | Test reaction on request when sent request is not answered | | |
| **References:** | RQ_002_6041 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6041_01 |

```
with {     IUT having established IKE_Security_Association
       and IUT having sent CREATE_CHILD_SA_request
       and IUT not having received CREATE_CHILD_SA_response
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request }
       then { IUT sends CREATE_CHILD_SA_response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6041_02** | | |
| **Summary:** | Test reaction on request when sent request is not answered | | |
| **References:** | RQ_002_6041 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6041_02 |

```
with {     IUT having established an IKE_Security_Association
       and IUT having sent INFORMATIONAL_request
       and IUT not having received INFORMATIONAL_response
     }
ensure that
     { when { IUT receives INFORMATIONAL_request }
       then { IUT sends INFORMATIONAL_response }
    }
```

### A.3.4.3.5 Request Internal Address

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6177_01** | | |
| **Summary:** | Test reaction on IKE_AUTH request with Configuration Payload | | |
| **References:** | RQ_002_6177, RQ_002_6178, RQ_002_6183, RQ_002_6462, RQ_002_6465 | | |
| **IUT Role** | Ipsec_gateway | **Test Case:** | TC_SEC_6177_01 |

```
with { IUT configured 'to expect IKE_AUTH request to include
                       the Configuration Payload'
     }
ensure that
     { when { IUT receives IKE_AUTH_request
                  containing (Configuration_payload
                              containing Configuration_Type
                                  set to 1 CFG_REQUEST
                         and containing (Configuration_Attribute
                                         containing Attribute_Type
                                             set to 8 INTERNAL_IP6_ADDRESS }
       then { IUT sends IKE_AUTH_response
                  containing (Configuration_Payload
                              containing Configuration_Type
                                  set to 2 CFG_REPLY
                         and containing (Configuration_Attribute
                                         containing Attribute_Type
                                             set to 8 INTERNAL_IP6_ADDRESS
                                       and containing Attribute_Value
                                             set to IPv6_Address)
                      before the Security_Association_payload }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6184_01** | | |
| **Summary:** | Test reaction on IKE_AUTH request with Configuration Payload | | |
| **References:** | RQ_002_6184, RQ_002_6462 | | |
| **IUT Role** | Ipsec_gateway | **Test Case:** | TC_SEC_6184_01 |

```
with { IUT configured 'to expect IKE_AUTH request to include
                       the Configuration Payload'
     }
ensure that
     { when { IUT receives IKE_AUTH_request
                  not containing (Configuration_payload
                                  containing Configuration_Type
                                      set to 1 CFG_REQUEST }
       then { IUT sends IKE_AUTH_response
                  containing (Notify_payload
                              containing Notify_Message_Type
                                  set to 37 FAILED_CP_REQUIRED) }
    }
```

### A.3.4.3.6 Retransmission Timers

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6030_01** | | |
| **Summary:** | Test reaction on repeated IKE_SA_INIT request | | |
| **References:** | RQ_002_6030, RQ_002_6046 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6030_01 |

```
with { ordered (     IUT having received IKE_SA_INIT_request
                   and IUT having sent IKE_SA_INIT_response
     }
ensure that
     { when { IUT receives previous IKE_SA_INIT_request -- i.e. the same as the
                                                        -- one that it has
                                                        -- already answered
            }
       then { IUT resends previous IKE_SA_INIT_response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6030_02** | | |
| **Summary:** | Test reaction on repeated IKE_AUTH request | | |
| **References:** | RQ_002_6030, RQ_002_6046 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6030_02 |

```
with { ordered (     IUT having received IKE_AUTH_request
                   and IUT having sent IKE_AUTH_response)
     }
ensure that
     { when { IUT receives previous IKE_AUTH_request    -- i.e. the same as the
                                                        -- one that it has
                                                        -- already answered
            }
       then { IUT resends previous IKE_AUTH_response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6030_03** | | |
| **Summary:** | Test reaction on repeated CREATE_CHILD_SA request | | |
| **References:** | RQ_002_6030, RQ_002_6046 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6030_03 |

```
with { ordered (     IUT having received CREATE_CHILD_SA_request
                   and IUT having sent CREATE_CHILD_SA_response)
     }
ensure that
     { when { IUT receives previous CREATE_CHILD_SA_request -- i.e. the same as
                                                            -- the one that it
                                                            -- has already
                                                            -- answered
            }
       then { IUT resends previous CREATE_CHILD_SA_response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6030_04** | | |
| **Summary:** | Test reaction on repeated INFORMATIONAL_request | | |
| **References:** | RQ_002_6030, RQ_002_6046 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6030_04 |

```
with { ordered (     IUT having received INFORMATIONAL_request
                  and IUT having sent INFORMATIONAL_response)
     }
ensure that
     { when { IUT receives previous INFORMATIONAL_request   -- i.e. the same as
                                                            -- the one that it
                                                            -- has already
                                                            -- answered
            }
       then { IUT resends previous INFORMATIONAL_response }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6033_01** | | |
| **Summary:** | Test resending of unanswered IKE_SA_INIT request | | |
| **References:** | RQ_002_6033, RQ_002_6045 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6033_01 |

```
with { IUT having sent IKE_SA_INIT_request
     }
ensure that
     { when { IUT receives no IKE_SA_INIT_response }
       then { IUT resends previous IKE_SA_INIT_request }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6033_02** | | |
| **Summary:** | Test resending of unanswered IKE_AUTH request | | |
| **References:** | RQ_002_6033, RQ_002_6045 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6033_02 |

```
with { IUT having sent IKE_AUTH_request
     }
ensure that
     { when { IUT receives no IKE_AUTH_response }
       then { IUT resends previous IKE_AUTH_request }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6033_03** | | |
| **Summary:** | Test resending of unanswered CREATE_CHILD_SA request | | |
| **References:** | RQ_002_6033, RQ_002_6045 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6033_03 |

```
with { IUT having sent CREATE_CHILD_SA_request
     }
ensure that
     { when { IUT receives no CREATE_CHILD_SA_response }
       then { IUT resends previous CREATE_CHILD_SA_request }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6033_04** | | |
| **Summary:** | Test resending of unanswered INFORMATIONAL_request | | |
| **References:** | RQ_002_6033, RQ_002_6045 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6033_04 |

```
with { IUT having sent INFORMATIONAL_request
     }
ensure that
     { when { IUT receives no INFORMATIONAL_response }
       then { IUT resends previous INFORMATIONAL_request }
    }
```

### A.3.4.3.7 Version Compatibility

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6065_01** | | |
| **Summary:** | Test reaction on IKE_SA_INIT request with major version > 2 | | |
| **References:** | RQ_002_6065, RQ_002_6066, RQ_002_6237 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6065_01 |

```
with { IUT ready to establish a Security_Association using IKEv2
     }
ensure that
     { when {     IUT receives IKE_SA_INIT_request
                      containing (IKE_Header
                                  containing Major_Version
                                      set to greater than 2) }
       then {     IUT discards IKE_SA_INIT_request
              and optionally (
                  IUT sends IKE_SA_INIT_response
                      containing (Notify_payload
                                  containing Notify_Message_Type
                                      set to 5 INVALID_MAJOR_VERSION) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6065_02** | | |
| **Summary:** | Test reaction on IKE_AUTH request with major version > 2 | | |
| **References:** | RQ_002_6065, RQ_002_6066, RQ_002_6237 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6065_02 |

```
with { ordered (     IUT having received IKE_SA_INIT_request
                 and IUT having sent IKE_SA_INIT_response)
     }
ensure that
     { when {     IUT receives IKE_AUTH_request
                      containing (IKE_Header
                                  containing Major_Version
                                      set to greater than 2) }
       then {     IUT discards IKE_AUTH_request
              and optionally (
                  IUT sends IKE_AUTH_response
                      containing (Notify_payload
                                  containing Notify_Message_Type
                                      set to 5 INVALID_MAJOR_VERSION) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6065_03** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with major version > 2 | | |
| **References:** | RQ_002_6065, RQ_002_6066, RQ_002_6237 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6065_03 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when {     IUT receives CREATE_CHILD_SA_request
                      containing (IKE_Header
                                      containing Major_Version
                                          set to greater than 2) }
       then {     IUT discards CREATE_CHILD_SA_request
              and optionally (
                  IUT sends CREATE_CHILD_SA_response
                      containing (Notify_payload
                                      containing Notify_Message_Type
                                          set to 5 INVALID_MAJOR_VERSION) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6065_04** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request with major version > 2 | | |
| **References:** | RQ_002_6065, RQ_002_6066, RQ_002_6237 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6065_04 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when {     IUT receives INFORMATIONAL_request
                      containing (IKE_Header
                                      containing Major_Version
                                          set to greater than 2 }
       then {     IUT discards INFORMATIONAL_request
              and optionally (
                  IUT sends INFORMATIONAL_response
                      containing (Notify_payload
                                      containing Notify_Message_Type
                                          set to 5 INVALID_MAJOR_VERSION) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6068_01** | | |
| **Summary:** | Test reaction on IKE_SA_INIT request with major version < 2 | | |
| **References:** | RQ_002_6068, RQ_002_6067, RQ_002_6069 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6068_01 |

```
with { IUT ready to establish a Security_Association using IKEv2
     }
ensure that
     { when { IUT receives IKE_SA_INIT_request
                  containing (IKE_Header
                                  containing Major_Version set to 1) }
       then { IUT sends IKE_SA_INIT_response
                  containing (IKE_Header
                                  containing Major_Version set to 1
                              and containing V_Bit set to 1) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6068_02** | | |
| **Summary:** | Test reaction on IKE_AUTH request with major version < 2 | | |
| **References:** | RQ_002_6068, RQ_002_6067, RQ_002_6069 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6068_02 |

```
with { ordered (     IUT having sent IKE_SA_INIT_request
                  and IUT having received IKE_SA_INIT_response)
     }
ensure that
     { when { IUT receives IKE_AUTH_request
                  containing (IKE_Header
                              containing Major_Version set to 1) }
       then { IUT sends IKE_AUTH_response
                  containing (IKE_Header
                              containing Major_Version set to 1
                            and containing V_Bit set to 1) }
     }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6068_03** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with major version < 2 | | |
| **References:** | RQ_002_6068, RQ_002_6067, RQ_002_6069 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6068_03 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                  containing (IKE_Header
                              containing Major_Version set to 1) }
       then { IUT sends CREATE_CHILD_SA_response
                  containing (IKE_Header
                              containing Major_Version set to 1
                            and containing V_Bit set to 1) }
     }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6068_04** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request with major version < 2 | | |
| **References:** | RQ_002_6068, RQ_002_6067, RQ_002_6069 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6068_04 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_request
                  containing (IKE_Header
                              containing Major_Version set to 1) }
       then { IUT sends INFORMATIONAL_response
                  containing (IKE_Header
                              containing Major_Version set to 1
                            and containing V_Bit set to 1) }
     }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6362_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with unrecognized payload | | |
| **References:** | RQ_002_6362, RQ_002_6255 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6362_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                  containing unrecognized (payload
                              containing C_Bit set to 1) }
       then { IUT sends CREATE_CHILD_SA_response
                  containing (Notify_payload
                              containing Notify_Message_Type
                                  set to 1 UNSUPPORTED_CRITICAL_PAYLOAD) }
     }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6362_02** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request with unrecognized payload | | |
| **References:** | RQ_002_6362, RQ_002_6255 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6362_02 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_request
                  containing unrecognized (payload
                              containing C_Bit set to 1) }
       then { IUT sends INFORMATIONAL_response
                  containing (Notify_payload
                              containing Notify_Message_Type
                                  set to 1 UNSUPPORTED_CRITICAL_PAYLOAD) }
     }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6073_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with unrecognized payload | | |
| **References:** | RQ_002_6073, RQ_002_6256 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6073_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                  containing unrecognized (payload
                              containing C_Bit set to 0) }
       then { IUT sends CREATE_CHILD_SA_response
                 not containing (Notify_payload
                              containing Notify_Message_Type
                                  set to 1 UNSUPPORTED_CRITICAL_PAYLOAD) }
     }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6073_02** | | |
| **Summary:** | Test reaction on INFORMATIONAL_request with unrecognized payload | | |
| **References:** | RQ_002_6073, RQ_002_6256 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6073_02 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives INFORMATIONAL_request
                  containing unrecognized (payload
                                containing C_Bit set to 0) }
       then { IUT sends INFORMATIONAL_response
              not containing (Notify_payload
                               containing Notify_Message_Type
                                   set to 1 UNSUPPORTED_CRITICAL_PAYLOAD) }
    }
```

## A.3.4.4 Security Parameter Negotiation

### A.3.4.4.1 Algorithm Negotiation

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6088_01** | | |
| **Summary:** | Test reaction on IKE_SA_INIT request with several SA proposal | | |
| **References:** | RQ_002_6088, RQ_002_6271 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6088_01 |

```
with { IUT ready to establish a Security_Association using IKEv2
     }
ensure that
     { when { IUT receives IKE_SA_INIT_request
                  containing (Security_Association_payload
                               containing at least 1 acceptable Proposal ) }
       then { IUT sends IKE_SA_INIT_response
                  containing (Security_Association_payload
                               containing 1 Proposal) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6088_02** | | |
| **Summary:** | Test reaction on IKE_AUTH request with several SA proposal | | |
| **References:** | RQ_002_6088, RQ_002_6271 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6088_02 |

```
with {     IUT having sent IKE_SA_INIT_request
       and IUT having received IKE_SA_INIT_response
     }
ensure that
     { when { IUT receives IKE_AUTH_request
                  containing (Security_Association_payload
                               containing at least 1 acceptable Proposal) }
       then { IUT sends IKE_AUTH_response
                  containing (Security_Association_payload
                               containing 1 Proposal) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6088_03** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with several SA proposal | | |
| **References:** | RQ_002_6088, RQ_002_6271 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6088_03 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                  containing (Security_Association_payload
                              containing at least 1 acceptable Proposal) }
       then { IUT sends CREATE_CHILD_SA_response
                  containing (Security_Association_payload
                              containing 1 Proposal) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6372_01** | | |
| **Summary:** | Test reaction on IKE_SA_INIT request with unacceptable SA proposal | | |
| **References:** | RQ_002_6372 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6372_01 |

```
with { IUT ready to establish a Security_Association using IKEv2
     }
ensure that
     { when { IUT receives IKE_SA_INIT_request
                  containing (Security_Association_payload
                              containing no acceptable Proposal) }
       then { IUT sends IKE_SA_INIT_response
                  containing (Notify_payload
                              containing Notify_Message_Type
                                  set to 14 NO_PROPOSAL_CHOSEN) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6372_02** | | |
| **Summary:** | Test reaction on IKE_AUTH request with unacceptable SA proposal | | |
| **References:** | RQ_002_6372 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6372_02 |

```
with {     IUT having sent IKE_SA_INIT_request
       and IUT having received IKE_SA_INIT_response
     }
ensure that
     { when { IUT receives IKE_AUTH_request
                  containing (Security_Association_payload
                              containing no acceptable Proposal) }
       then { IUT sends IKE_AUTH_response
                  containing (Notify_payload
                              containing Notify_Message_Type
                                  set to 14 NO_PROPOSAL_CHOSEN) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6372_03** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with unacceptable SA proposal | | |
| **References:** | RQ_002_6372 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6372_03 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                    containing (Security_Association_payload
                                containing no acceptable Proposal) }
       then { IUT sends CREATE_CHILD_SA_response
                    containing (Notify_payload
                                containing Notify_Message_Type
                                    set to 14 NO_PROPOSAL_CHOSEN) }
     }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6373_01** | | |
| **Summary:** | Test reaction on IKE_SA_INIT request with invalid Diffie-Hellman value | | |
| **References:** | RQ_002_6373, RQ_002_6306 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6373_01 |

```
with { IUT ready to establish a Security_Association using IKEv2
     }
ensure that
     { when { IUT receives IKE_SA_INIT_request
                    containing (Key_Exchange_payload
                                containing an invalid DH_Group_number) }
       then { IUT sends IKE_SA_INIT_response
                    containing (Notify_payload
                                containing Notify_Message_Type
                                    set to 17 INVALID_KE_PAYLOAD) }
     }
```

### A.3.4.4.2 Cookies

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6081_01** | | |
| **Summary:** | Test reaction on IKE_SA_INIT response with COOKIE Notify payload | | |
| **References:** | RQ_002_6081, RQ_002_6080, RQ_002_6391 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6081_01 |

```
with { IUT having sent IKE_SA_INIT_request
     }
ensure that
     { when { IUT receives IKE_SA_INIT_response
                    containing (Notify_payload
                                containing Notify_Message_Type
                                    set to 16390 COOKIE
                            and containing (Notification_Data
                                             containing 'Cookie data') }
       then { IUT sends IKE_SA_INIT_request
                    containing (Notify_payload
                                containing Notify_Message_Type
                                    set to 16390 COOKIE
                            and containing Notification_Data
                                    set to Notification_Data
                                           received in IKE_SA_INIT_response)
              and containing 'all other payloads from initial
                                request unchanged' }
     }
```

### A.3.4.4.3 Rekeying

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6101_01** | | |
| **Summary:** | Test of generating CREATE_CHILD_SA request for rekeying of child SA | | |
| **References:** | RQ_002_6101, RQ_002_6172, RQ_002_6173, RQ_002_6397 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6101_01 |

```
with {     IUT having established an IKE_Security_Association
       and IUT having established a CHILD_SA
       and IUT 'having detected that the lifetime of the CHILD_SA
                is about to expire'
       and IUT 'able to rekey CHILD_SA within IKE_SA'
     }
ensure that
     { when { IUT is requested to send CREATE_CHILD_SA_request }
       then { IUT sends CREATE_CHILD_SA_request
                  containing (Notify_payload
                              containing Notify_Message_Type
                                  set to 16393 REKEY_SA) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6102_01** | | |
| **Summary:** | Test of deletion of old CREATE_CHILD_SA after rekeying | | |
| **References:** | RQ_002_6102 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6102_01 |

```
with {     IUT having established an IKE_Security_Association
       and IUT having established a CHILD_SA
       and IUT 'having detected that the lifetime of the CHILD_SA
                was about to expire'
       and IUT having sent CREATE_CHILD_SA_request 'for rekeying'
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_response }
       then { IUT sends INFORMATIONAL_request
                  containing (Delete_payload
                              containing Security_Parameters_Index
                              indicating CHILD_SA 'to be deleted') }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6103_01** | | |
| **Summary:** | Test of generating CREATE_CHILD_SA request for rekeying of IKE SA | | |
| **References:** | RQ_002_6103 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6103_01 |

```
with {     IUT having established an IKE_Security_Association
       and IUT having established a CHILD_SA
       and IUT 'having detected that the lifetime of the IKE_SA
                was about to expire'
     }
ensure that
     { when { IUT is requested to send CREATE_CHILD_SA_request }
       then { IUT sends CREATE_CHILD_SA_request
              not containing Traffic_Selector_payload_initiator
              and not containing Traffic_Selector_payload_responder }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6105_01** | | |
| **Summary:** | Test of deletion of old IKE_SA after rekeying | | |
| **References:** | RQ_002_6105 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6105_01 |

```
with {     IUT having established an IKE_Security_Association
       and IUT having established a CHILD_SA
       and IUT 'having detected that the lifetime of the CHILD_SA
                was about to expire'
       and IUT 'has rekeyed IKE_SA'
     }
ensure that
     { when { IUT is requested to send INFORMATIONAL_request }
       then { IUT sends INFORMATIONAL_request
                  containing (Delete_payload
                              containing Security_Parameters_Index
                              indicating IKE_Security_Association
                                         'to be deleted') }
    }
```

### A.3.4.4.4 Traffic Selector Negotiation

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6123_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with acceptable and unacceptable traffic selectors | | |
| **References:** | RQ_002_6123 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6123_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                  containing (Traffic_Selector_payload_initiator
                              containing first
                                      and acceptable Traffic_Selector
                         and containing next
                                     and unacceptable Traffic_Selector)
            and containing (Traffic_Selector_payload_responder
                              containing first
                                      and acceptable Traffic_Selector
                         and containing next
                                     and unacceptable Traffic_Selector) }
       then { IUT sends CREATE_CHILD_SA_response
                  containing (Traffic_Selector_payload_initiator
                              containing acceptable Traffic_Selector
                                received in CREATE_CHILD_SA_request)
            and containing (Traffic_Selector_payload_responder
                              containing acceptable Traffic_Selector
                                received in CREATE_CHILD_SA_request) }
    }
```

| Test Purpose | | | |
|---|---|---|---|
| **Identifier:** | **TP_SEC_6125_01** | | |
| **Summary:** | Test reaction on CREATE_CHILD_SA request with acceptable and unacceptable traffic selectors | | |
| **References:** | RQ_002_6125, RQ_002_6383 | | |
| **IUT Role** | Host | **Test Case:** | TC_SEC_6125_01 |

```
with { IUT having established an IKE_Security_Association
     }
ensure that
     { when { IUT receives CREATE_CHILD_SA_request
                  containing (Traffic_Selector_payload_initiator
                              containing Traffic_Selector
                              indicating 'a range of parameters of which
                                          only a subset is acceptable')
              and containing (Traffic_Selector_payload_responder
                              containing Traffic_Selector
                                  set to 'a range of parameters of which
                                          only a subset is acceptable') }
       then { IUT sends CREATE_CHILD_SA_response
                  containing (Traffic_Selector_payload_initiator
                              containing Traffic_Selector
                                  set to 'acceptable subset of range'
                                         received in CREATE_CHILD_SA_request)
              and containing (Traffic_Selector_payload_responder
                              containing Traffic_Selector
                                  set to 'acceptable subset of range'
                                         received in CREATE_CHILD_SA_request)
              and optionally (
                  containing (Notify_payload
                              containing Notify_Message_Type
                                  set to 16386 ADDITIONAL_TS_POSSIBLE) }
    }
```

# Annex B (informative): Bibliography

IETF RFC 4301: "Security Architecture for the Internet Protocol".

IETF RFC 4302: "IP Authentication Header".

IETF RFC 4303: "IP Encapsulating Security Payload (ESP)".

IETF RFC 4306: "Internet Key Exchange (IKEv2) Protocol".

# History

| Document history | | |
|---|---|---|
| V1.1.1 | April 2007 | Publication |
| | | |
| | | |
| | | |
| | | |