Understanding IPSec IKEv1 negotiation on Wireshark

Rodrigo_Albuque

MVP

on 01-May-2019 00:54

Related Articles:

Understanding IPSec IKEv2 negotiation on Wireshark

1. The Big Picture

First 6 Identity Protection (Main Mode) messages negotiate security parameters to protect the next 3 messages (Quick Mode) and whatever is negotiated in Phase 2 is used to protect production traffic (ESP or AH, normally ESP for site-site VPN).

We call first 6 messages Phase 1 and last 3 messages as Phase 2.

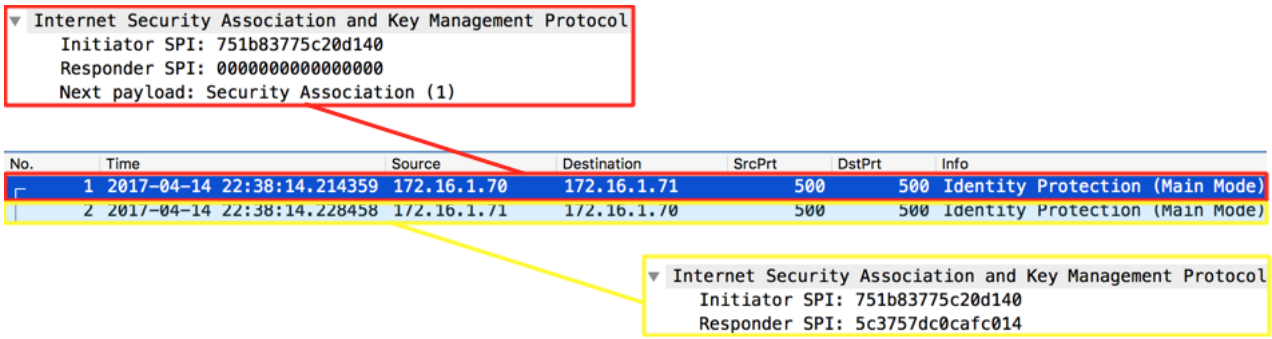
No.	Time	Source	Destination	SrcPrt	DstPrt	Info
1	2017-04-14 22:38:14.214359	172.16.1.70	172.16.1.71	500	500	Identity Protection (Main Mode)
2	2017-04-14 22:38:14.228458	172.16.1.71	172.16.1.70	500	500	Identity Protection (Main Mode)
3	2017-04-14 22:38:14.246521	172.16.1.70	172.16.1.71	500	500	Identity Protection (Main Mode)
4	2017-04-14 22:38:14.250607	172.16.1.71	172.16.1.70	500	500	Identity Protection (Main Mode)
5	2017-04-14 22:38:14.263722	172.16.1.70	172.16.1.71	500	500	Identity Protection (Main Mode)
6	2017-04-14 22:38:14.264785	172.16.1.71	172.16.1.70	500	500	Identity Protection (Main Mode)
7	2017-04-14 22:38:14.281969	172.16.1.70	172.16.1.71	500	500	Quick Mode
8	2017-04-14 22:38:14.282573	172.16.1.71	172.16.1.70	500	500	Quick Mode
9	2017-04-14 22:38:14.445523	172.16.1.70	172.16.1.71	500	500	Quick Mode

Sample pcap: IPSEC-tunnel-capture-1.pcap (for instructions on how to decrypt it just go to website where I got this sample capture: http://ruwanindikaprasanna.blogspot.com/2017/04/ipsec-capture-with-decryption.html)

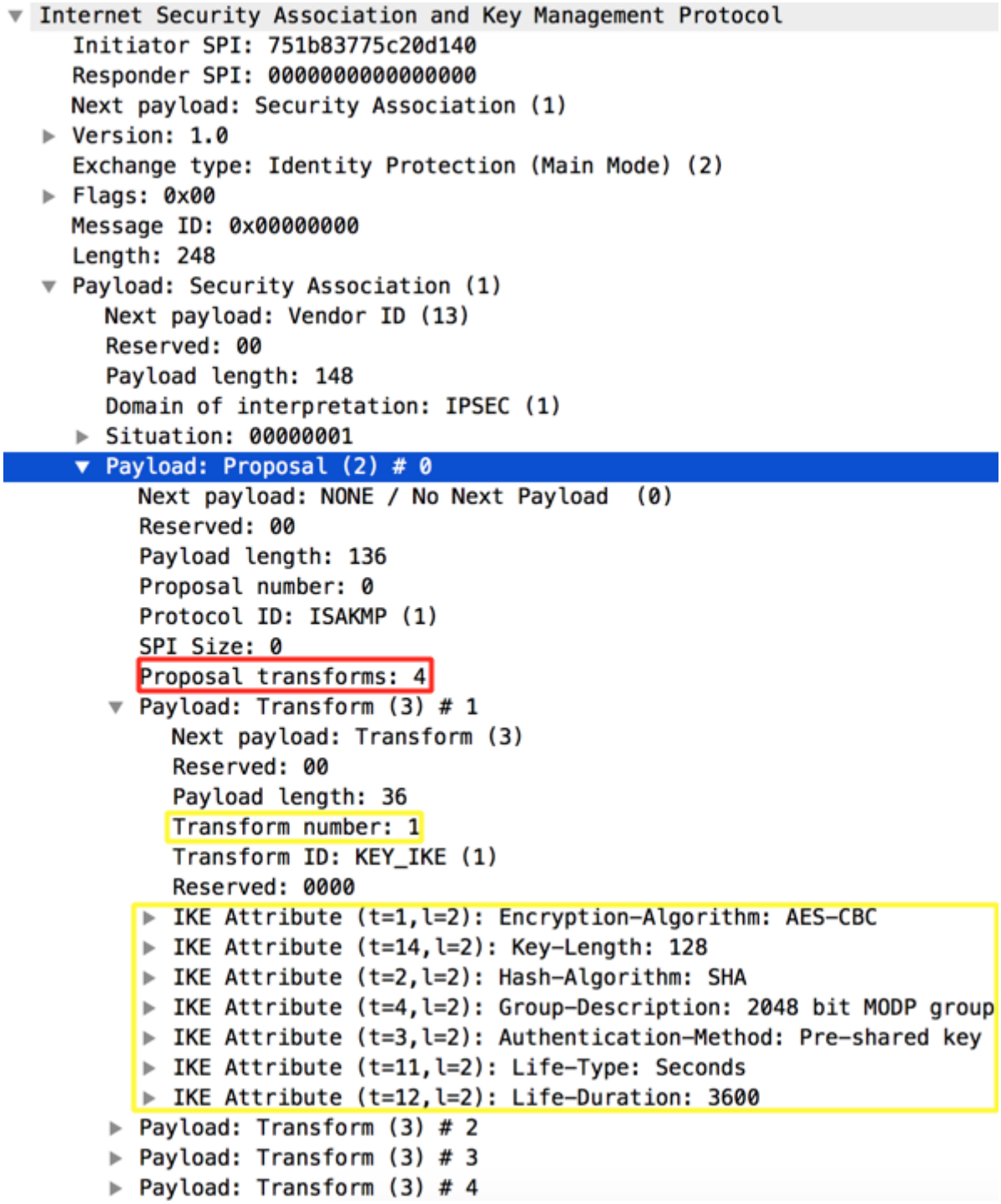
2. Phase 1

2.1 Policy Negotiation

Both peers add a unique SPI just to uniquely identify each side's Security Association (SA):



In **frame #1**, the Initiator (**.70**) sends a set of Proposals containing a set of security parameters (**Transforms**) that Responder (**.71**) can pick if it matches its local policies:

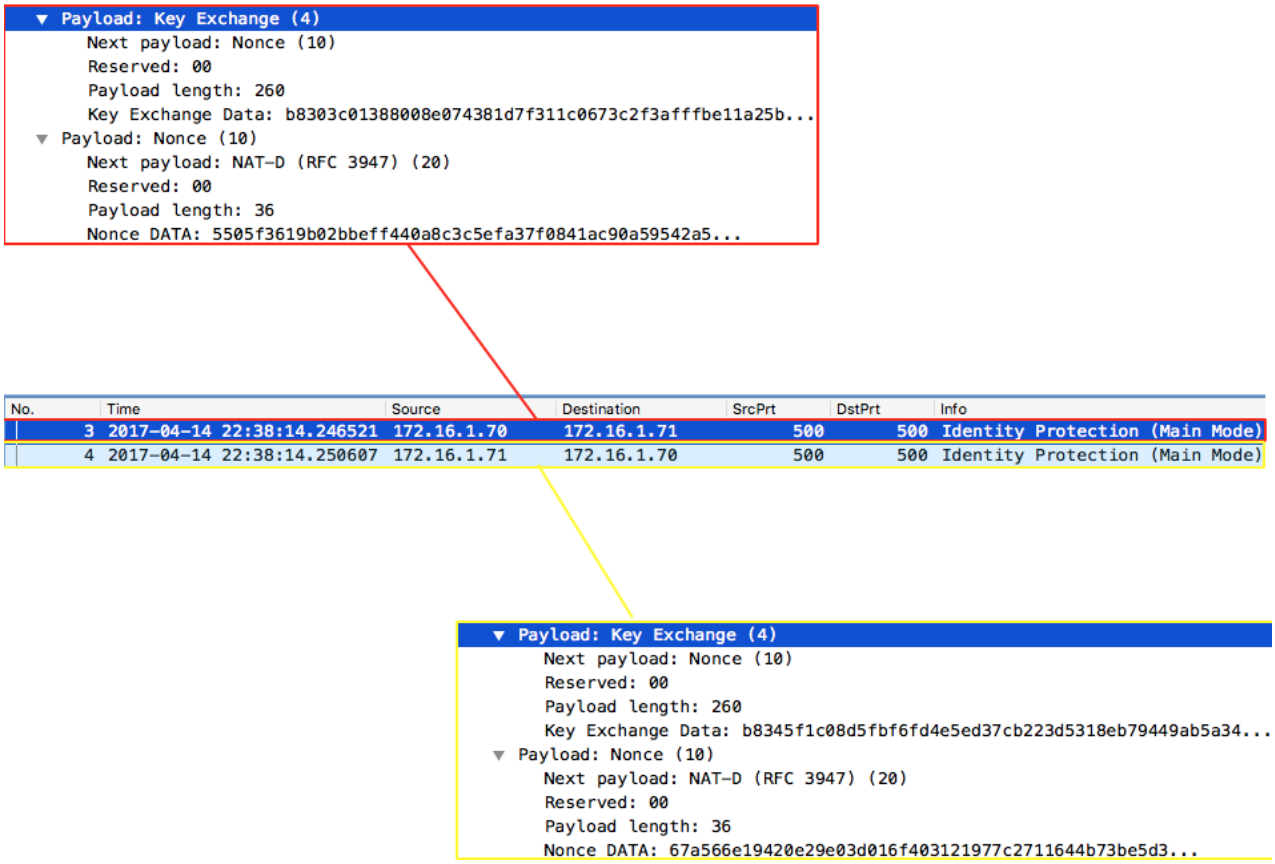


Fair enough, in frame #2 the Responder (**.71**) picks one of the **Transforms**:

```
▼ Payload: Security Association (1)
    Next payload: Vendor ID (13)
    Reserved: 00
    Payload length: 56
    Domain of interpretation: IPSEC (1)
  ▶ Situation: 00000001
  ▼ Payload: Proposal (2) # 0
      Next payload: NONE / No Next Payload  (0)
      Reserved: 00
      Payload length: 44
      Proposal number: 0
      Protocol ID: ISAKMP (1)
      SPI Size: 0
      Proposal transforms: 1
    ▼ Payload: Transform (3) # 1
        Next payload: NONE / No Next Payload  (0)
        Reserved: 00
        Payload length: 36
        Transform number: 1
        Transform ID: KEY_IKE (1)
        Reserved: 0000
      ▶ IKE Attribute (t=1,l=2): Encryption-Algorithm: AES-CBC
      ▶ IKE Attribute (t=14,l=2): Key-Length: 128
      ▶ IKE Attribute (t=2,l=2): Hash-Algorithm: SHA
      ▶ IKE Attribute (t=4,l=2): Group-Description: 2048 bit MODP group
      ▶ IKE Attribute (t=3,l=2): Authentication-Method: Pre-shared key
      ▶ IKE Attribute (t=11,l=2): Life-Type: Seconds
      ▶ IKE Attribute (t=12,l=2): Life-Duration: 3600
```

2.2 DH Key Exchange

Then, next 2 Identity Protection packets both peers exchange Diffie-Hellman public key values and nonces (random numbers) which will then allow both peers to agree on a shared secret key:



With DH public key value and the nonce both peers will generate a seed key called SKEYID.

A further 3 session keys will be generated using this seed key for different purposes:

SKEYID_d (d for derivative): not used by Phase 1. It is used as seed key for Phase2 keys, i.e. seed key for production traffic keys in Plain English.

SKEYID_a (a for authentication): this key is used to protect message integrity in every subsequent packets as soon as both peers are authenticated (peers will authenticate each other in next 2 packets). Yes, I know, we verify the integrity by using a hash but throwing a key into a hash adds stronger security to hash and it's called HMAC.

SKEYID_e (e for encryption): you'll see that the next 2 packets are also encrypted. As selected encryption algorithm for this phase was AES-CBC (128-bits) then we use AES with this key to symmetrically encrypt further data.

Nonce is just to protect against replay attacks by adding some randomness to key generation

2.3 Authentication

The purpose of this exchange is to confirm each other's identity. If we said we're going to do this using pre-shared keys then verification consists of checking whether both sides has the same pre-shared key. If it is RSA certificate then peers exchange RSA certificates and assuming the CA that signed each side is trusted then verification complete successfully.

In our case, this is done via pre-shared keys:

No.	Time	Source	Destination	SrcPrt	DstPrt	Info
5	2017-04-14 22:38:14.263722	172.16.1.70	172.16.1.71	500	500	Identity Protection (Main Mode)
6	2017-04-14 22:38:14.264785	172.16.1.71	172.16.1.70	500	500	Identity Protection (Main Mode)

In packet #5 the Initiator sends a hash generated using pre-shared key set as key material so that only those who possess pre-master key can do it:

```
▼ Internet Security Association and Key Management Protocol
    Initiator SPI: 751b83775c20d140
    Responder SPI: 5c3757dc0cafc014
    Next payload: Identification (5)
  ▶ Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  ▼ Flags: 0x01
      .... ...1 = Encryption: Encrypted
      .... ..0. = Commit: No commit
      .... .0.. = Authentication: No authentication
    Message ID: 0x00000000
    Length: 108
  ▼ Encrypted Data (80 bytes)
    ▼ Payload: Identification (5)
        Next payload: Hash (8)
        Reserved: 00
        Payload length: 27
        ID type: FQDN (2)
        Protocol ID: Unused
        Port: Unused
      ▶ Identification Data:moon.strongswan.org
    ▼ Payload: Hash (8)
        Next payload: Notification (11)
        Reserved: 00
        Payload length: 24
        Hash DATA: 14ff218df52306c134b5431bd88e3a2809fee996
    ▼ Payload: Notification (11)
        Next payload: NONE / No Next Payload  (0)
        Reserved: 00
        Payload length: 28
        Domain of interpretation: IPSEC (1)
        Protocol ID: ISAKMP (1)
        SPI Size: 16
        Notify Message Type: INITIAL-CONTACT (24578)
        SPI: 751b83775c20d1405c3757dc0cafc014
        Notification DATA: <MISSING>
      Extra data: 00
```

The responder performs the same calculation and confirms the hash is correct.

Responder also sends a similar packet back to Initiator in frame #6 but I skipped for brevity.

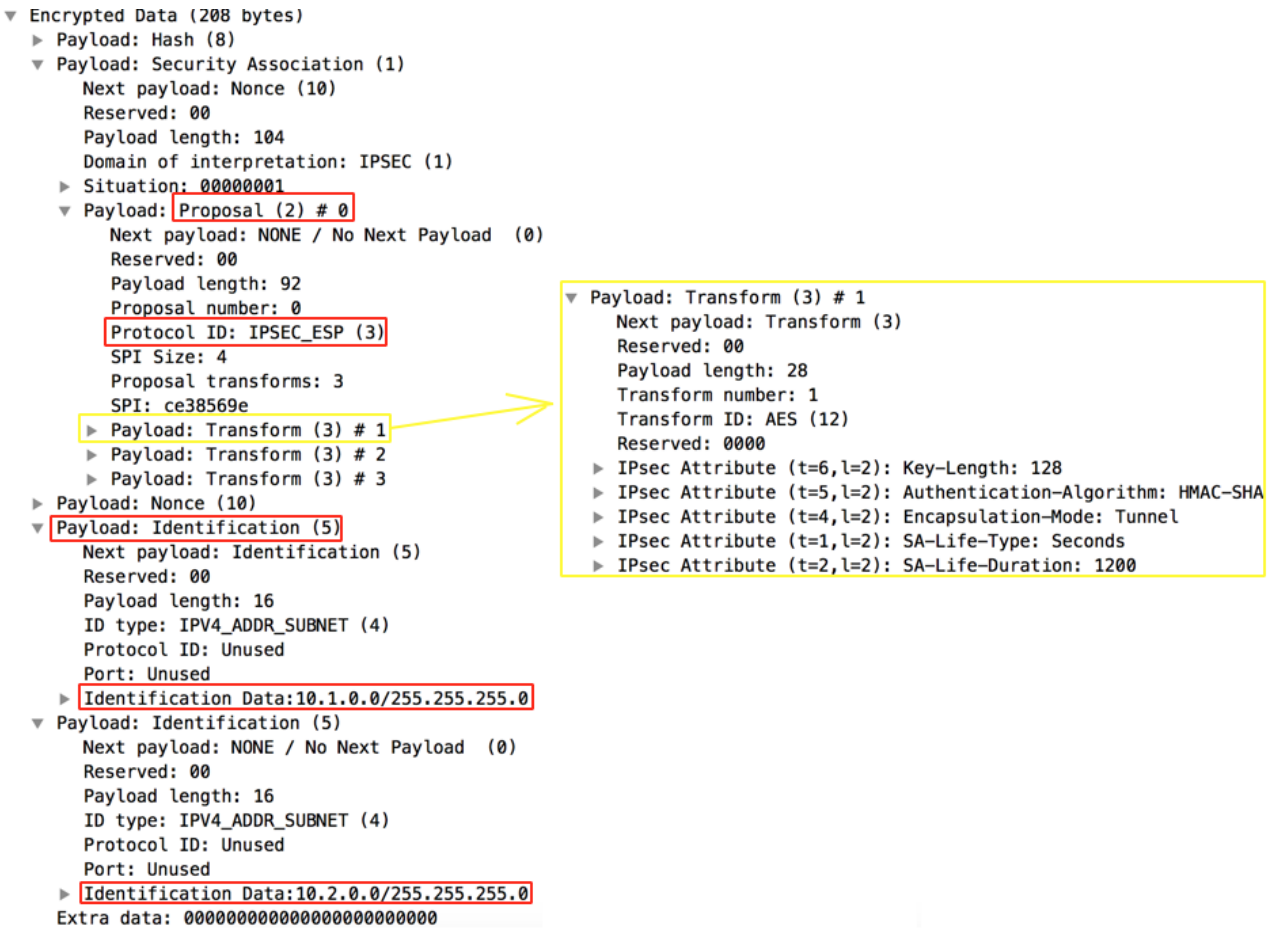
Now we're ready for Phase 2.

3. Phase 2

The purpose of this phase is to establish the security parameters that will be used for production traffic (IPSec SA):

No.	Time	Source	Destination	SrcPrt	DstPrt	Info
7	2017-04-14 22:38:14.281969	172.16.1.70	172.16.1.71	500	500	Quick Mode
8	2017-04-14 22:38:14.282573	172.16.1.71	172.16.1.70	500	500	Quick Mode
9	2017-04-14 22:38:14.445523	172.16.1.70	172.16.1.71	500	500	Quick Mode

Now, Initiator sends its proposals to negotiate the security parameters for production traffic as mentioned (the highlighted yellow proposal is just a sample as the rest is collapsed - **this is frame #7**):

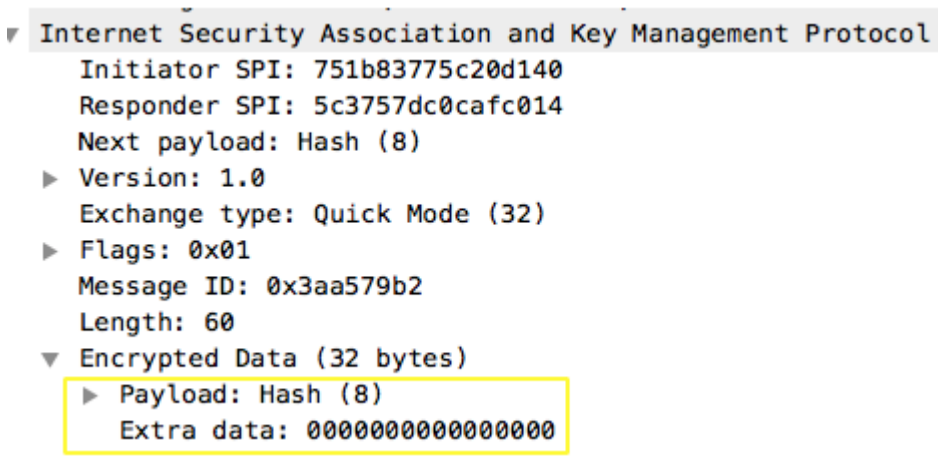


Note: Identification payload carries source and destination tunnel IP addresses and if this doesn't match what is configured on both peers then IPSec negotiation will not proceed.

Then, in frame #8 we see that Responder picked one of the Proposals:

```
▼ Payload: Security Association (1)
    Next payload: Nonce (10)
    Reserved: 00
    Payload length: 52
    Domain of interpretation: IPSEC (1)
  ▶ Situation: 00000001
  ▼ Payload: Proposal (2) # 0
      Next payload: NONE / No Next Payload  (0)
      Reserved: 00
      Payload length: 40
      Proposal number: 0
      Protocol ID: IPSEC_ESP (3)
      SPI Size: 4
      Proposal transforms: 1
      SPI: c04af751
    ▼ Payload: Transform (3) # 1
        Next payload: NONE / No Next Payload  (0)
        Reserved: 00
        Payload length: 28
        Transform number: 1
        Transform ID: AES (12)
        Reserved: 0000
      ▶ IPsec Attribute (t=6,l=2): Key-Length: 128
      ▶ IPsec Attribute (t=5,l=2): Authentication-Algorithm: HMAC-SHA
      ▶ IPsec Attribute (t=4,l=2): Encapsulation-Mode: Tunnel
      ▶ IPsec Attribute (t=1,l=2): SA-Life-Type: Seconds
      ▶ IPsec Attribute (t=2,l=2): SA-Life-Duration: 1200
▶ Payload: Nonce (10)
▼ Payload: Identification (5)
    Next payload: Identification (5)
    Reserved: 00
    Payload length: 16
    ID type: IPV4_ADDR_SUBNET (4)
    Protocol ID: Unused
    Port: Unused
  ▶ Identification Data:10.1.0.0/255.255.255.0
▼ Payload: Identification (5)
    Next payload: NONE / No Next Payload  (0)
    Reserved: 00
    Payload length: 16
    ID type: IPV4_ADDR_SUBNET (4)
    Protocol ID: Unused
    Port: Unused
  ▶ Identification Data:10.2.0.0/255.255.255.0
  Extra data: 00000000000000000000000000000000
```

Frame #9 is just an ACK to the picked proposal confirming that Initiator accepted it:



I just highlighted the Hash here to reinforce the fact that since both peers were authenticated in Phase 1, all subsequent messages are authenticated and a new hash (HMAC) is generated for each packet.

Application Delivery **Security**

BIG-IP ipsec LTM

1 Kudo

Version history

Last update:

01-May-2019 00:54

Updated by:

Rodrigo_Albuque

Contributors

Rodrigo_Albuque

Technical Forum
Technical Articles
CrowdSRC
Community Guidelines
DevCentral EULA
Get a Developer Lab License
Become a DevCentral MVP

Glossary
Customer Stories
Webinars
Free Online Courses
F5 Certification
LearnF5 Training

Professional Services
Create a Service Request
Software Downloads

F5 PARTNERS +

Find a Reseller Partner
Technology Alliances

Become an F5 Partner
Login to Partner Central

CONNECT WITH DEVCENTRAL

Trademarks Policies Privacy California Privacy
Do Not Sell My Personal Information Cookie Preferences