Help

Search all content

Turn on suggestions

DevCentral > Technical Articles > Understanding IPSec IKEv2 negotiation on Wireshark

Options

Understanding IPSec IKEv2 negotiation on Wireshark

Rodrigo_Albuque

MVP

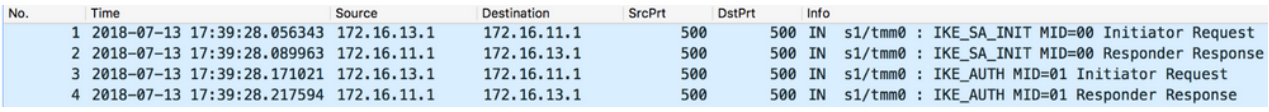
on 22-Jul-2019 05:08

Related Articles:

Understanding IPSec IKEv1 negotiation on Wireshark

1 The Big Picture

There are just 4 messages:



```
No.     Time                        Source        Destination   SrcPrt  DstPrt  Info
      1 2018-07-13 17:39:28.056343  172.16.13.1   172.16.11.1      500     500  IN  s1/tmm0 : IKE_SA_INIT MID=00 Initiator Request
      2 2018-07-13 17:39:28.089963  172.16.11.1   172.16.13.1      500     500  IN  s1/tmm0 : IKE_SA_INIT MID=00 Responder Response
      3 2018-07-13 17:39:28.171021  172.16.13.1   172.16.11.1      500     500  IN  s1/tmm0 : IKE_AUTH MID=01 Initiator Request
      4 2018-07-13 17:39:28.217594  172.16.11.1   172.16.13.1      500     500  IN  s1/tmm0 : IKE_AUTH MID=01 Responder Response
```

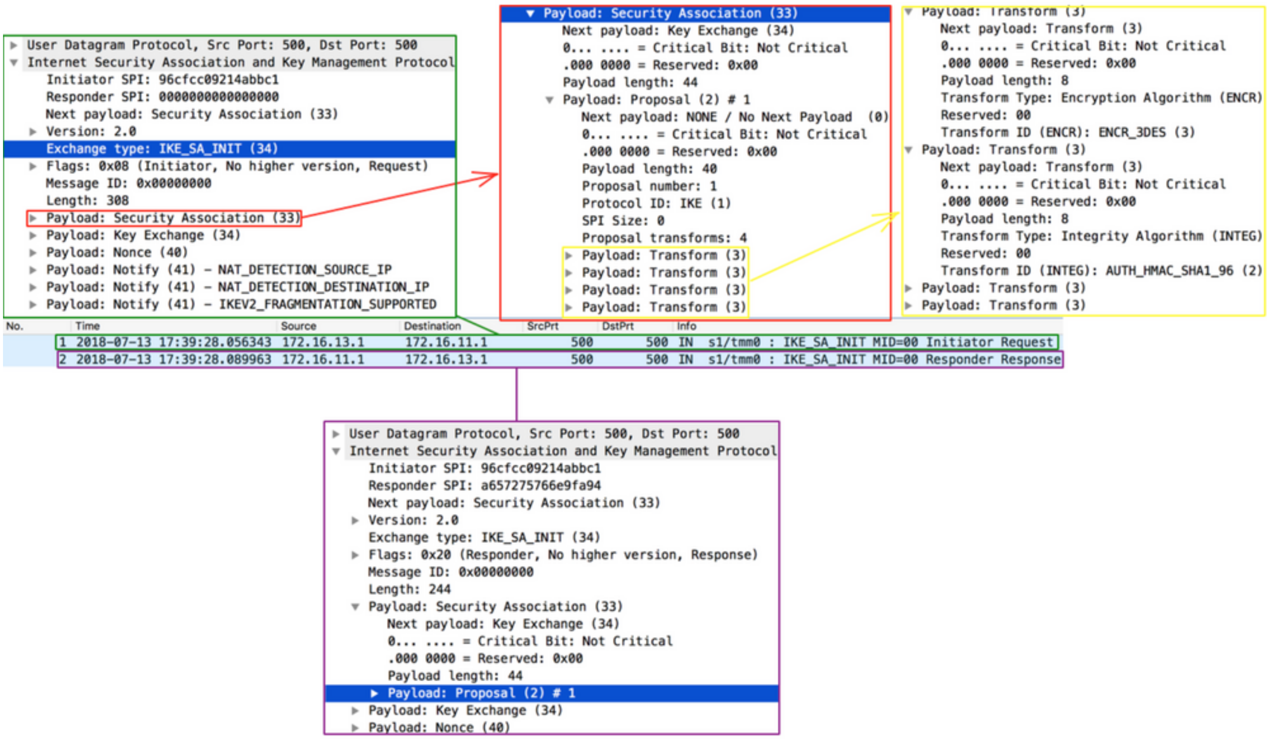
Summary:

- **IKE_SA_INIT**: negotiate security parameters to protect the next 2 messages (IKE_AUTH)
- Also creates a seed key (known as SKEYSEED) where further keys are produced:
- SK_e (encryption): computed for each direction (one for outbound and one for inbound) to encrypt IKE_AUTH messages
- SK_a (authentication): computed for each direction (one for outbound and one for inbound) to hash (using HMAC) IKE_AUTH messages

- SK_d (derivation): handed to IPSec to generate encryption and optionally authentication keys for production traffic
- **IKE_AUTH**: negotiates security parameters to protect production traffic (CHILD_SA)
- More specifically, the IPSec protocol used (ESP or AH - typically ESP as AH doesn't support encryption), the Encryption algorithm (AES128? AES256?) and Authentication algorithm (HMAC_SHA256? HMAC_SHA384?).

2 IKE_SA_INIT

First the Initiator sends a **Security Association** —> **Proposal** —> **Transform**, **Transform**... payloads which contains the required security settings to protect **IKE_AUTH** phase as well as to generate the seed key (**SK_d**) for production traffic (child SA):



```
▶ User Datagram Protocol, Src Port: 500, Dst Port: 500
▼ Internet Security Association and Key Management Protocol
    Initiator SPI: 96cfcc09214abbc1
    Responder SPI: 0000000000000000
    Next payload: Security Association (33)
  ▶ Version: 2.0
    Exchange type: IKE_SA_INIT (34)
  ▶ Flags: 0x08 (Initiator, No higher version, Request)
    Message ID: 0x00000000
    Length: 308
  ▶ Payload: Security Association (33)
  ▶ Payload: Key Exchange (34)
  ▶ Payload: Nonce (40)
  ▶ Payload: Notify (41) - NAT_DETECTION_SOURCE_IP
  ▶ Payload: Notify (41) - NAT_DETECTION_DESTINATION_IP
  ▶ Payload: Notify (41) - IKEV2_FRAGMENTATION_SUPPORTED
```

```
▼ Payload: Security Association (33)
    Next payload: Key Exchange (34)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 44
  ▼ Payload: Proposal (2) # 1
      Next payload: NONE / No Next Payload  (0)
      0... .... = Critical Bit: Not Critical
      .000 0000 = Reserved: 0x00
      Payload length: 40
      Proposal number: 1
      Protocol ID: IKE (1)
      SPI Size: 0
      Proposal transforms: 4
    ▶ Payload: Transform (3)
    ▶ Payload: Transform (3)
    ▶ Payload: Transform (3)
    ▶ Payload: Transform (3)
```

```
▼ Payload: Transform (3)
    Next payload: Transform (3)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 8
    Transform Type: Encryption Algorithm (ENCR)
    Reserved: 00
    Transform ID (ENCR): ENCR_3DES (3)
▼ Payload: Transform (3)
    Next payload: Transform (3)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 8
    Transform Type: Integrity Algorithm (INTEG)
    Reserved: 00
    Transform ID (INTEG): AUTH_HMAC_SHA1_96 (2)
▶ Payload: Transform (3)
▶ Payload: Transform (3)
```

```
No.  Time                        Source       Destination  SrcPrt  DstPrt  Info
1    2018-07-13 17:39:28.056343  172.16.13.1  172.16.11.1  500     500 IN  s1/tmm0 : IKE_SA_INIT MID=00 Initiator Request
2    2018-07-13 17:39:28.089963  172.16.11.1  172.16.13.1  500     500 IN  s1/tmm0 : IKE_SA_INIT MID=00 Responder Response
```

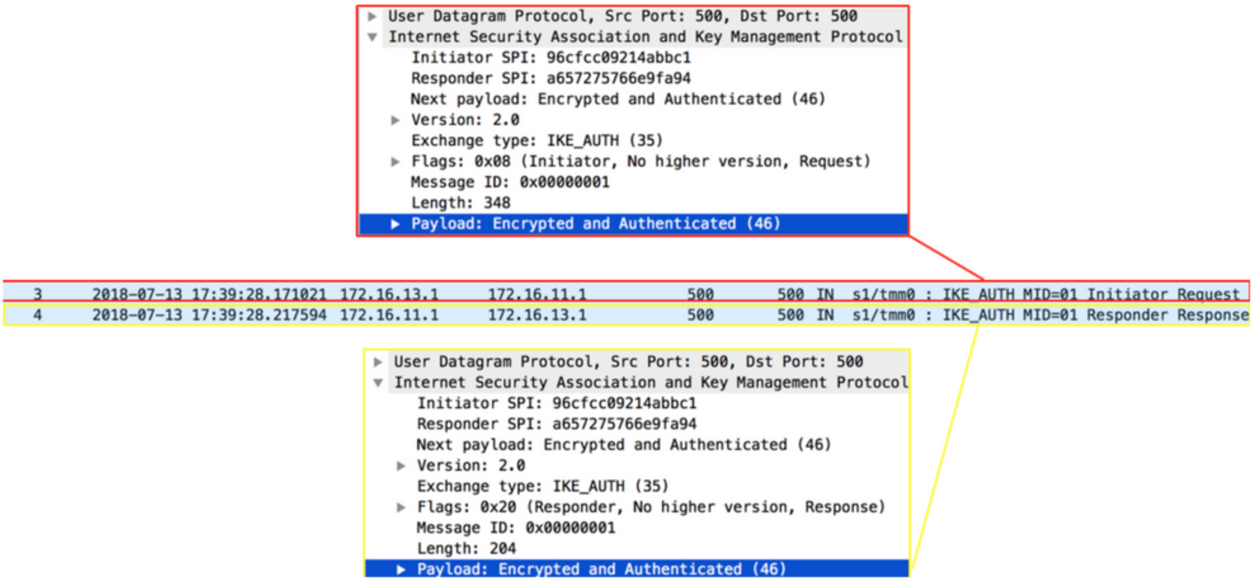
```
▶ User Datagram Protocol, Src Port: 500, Dst Port: 500
▼ Internet Security Association and Key Management Protocol
    Initiator SPI: 96cfcc09214abbc1
    Responder SPI: a657275766e9fa94
    Next payload: Security Association (33)
  ▶ Version: 2.0
    Exchange type: IKE_SA_INIT (34)
  ▶ Flags: 0x20 (Responder, No higher version, Response)
    Message ID: 0x00000000
    Length: 244
  ▼ Payload: Security Association (33)
      Next payload: Key Exchange (34)
      0... .... = Critical Bit: Not Critical
      .000 0000 = Reserved: 0x00
      Payload length: 44
    ▶ Payload: Proposal (2) # 1
  ▶ Payload: Key Exchange (34)
  ▶ Payload: Nonce (40)
```

In this case here the Initiator only sent one option for Encryption, Integrity, Pseudo-Random Function (PRF) and Diffie Hellman group so there are only 4 corresponding transforms but there could be more.

Responder picked the 4 available security options also confirmed in **Security Association** —> **Proposal** —> **Transform**, **Transform**... payloads as seen above.

3 IKE_AUTH

These are immediately applied to next 2 **IKE_AUTH** messages as seen below:



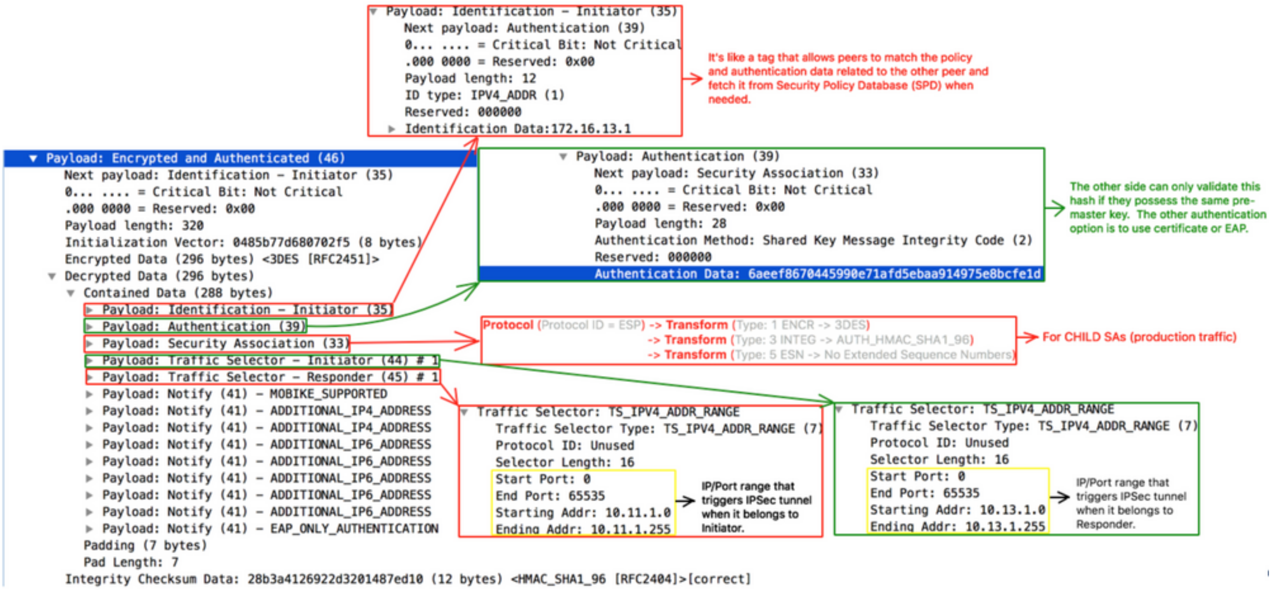
```
▶ User Datagram Protocol, Src Port: 500, Dst Port: 500
▼ Internet Security Association and Key Management Protocol
    Initiator SPI: 96cfcc09214abbc1
    Responder SPI: a657275766e9fa94
    Next payload: Encrypted and Authenticated (46)
  ▶ Version: 2.0
    Exchange type: IKE_AUTH (35)
  ▶ Flags: 0x08 (Initiator, No higher version, Request)
    Message ID: 0x00000001
    Length: 348
  ▶ Payload: Encrypted and Authenticated (46)
```

```
3    2018-07-13 17:39:28.171021  172.16.13.1    172.16.11.1     500     500  IN   s1/tmm0 : IKE_AUTH MID=01 Initiator Request
4    2018-07-13 17:39:28.217594  172.16.11.1    172.16.13.1     500     500  IN   s1/tmm0 : IKE_AUTH MID=01 Responder Response
```

```
▶ User Datagram Protocol, Src Port: 500, Dst Port: 500
▼ Internet Security Association and Key Management Protocol
    Initiator SPI: 96cfcc09214abbc1
    Responder SPI: a657275766e9fa94
    Next payload: Encrypted and Authenticated (46)
  ▶ Version: 2.0
    Exchange type: IKE_AUTH (35)
  ▶ Flags: 0x20 (Responder, No higher version, Response)
    Message ID: 0x00000001
    Length: 204
  ▶ Payload: Encrypted and Authenticated (46)
```

The above payload is Encrypted using SK_e and Integrity-protected using SK_a (these keys are different for each direction).

The first **IKE_AUTH** message negotiates the security parameters for production traffic (child SAs), authenticates each side and informs what is the source/destination IP/Port that is supposed to go through IPSec tunnel:



```
▼ Payload: Encrypted and Authenticated (46)
    Next payload: Identification - Initiator (35)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 320
    Initialization Vector: 0485b77d680702f5 (8 bytes)
    Encrypted Data (296 bytes) <3DES [RFC2451]>
  ▼ Decrypted Data (296 bytes)
    ▼ Contained Data (288 bytes)
      ▶ Payload: Identification - Initiator (35)
      ▶ Payload: Authentication (39)
      ▶ Payload: Security Association (33)
      ▶ Payload: Traffic Selector - Initiator (44) # 1
      ▶ Payload: Traffic Selector - Responder (45) # 1
      ▶ Payload: Notify (41) - MOBIKE_SUPPORTED
      ▶ Payload: Notify (41) - ADDITIONAL_IP4_ADDRESS
      ▶ Payload: Notify (41) - ADDITIONAL_IP4_ADDRESS
      ▶ Payload: Notify (41) - ADDITIONAL_IP6_ADDRESS
      ▶ Payload: Notify (41) - ADDITIONAL_IP6_ADDRESS
      ▶ Payload: Notify (41) - ADDITIONAL_IP6_ADDRESS
      ▶ Payload: Notify (41) - ADDITIONAL_IP6_ADDRESS
      ▶ Payload: Notify (41) - ADDITIONAL_IP6_ADDRESS
      ▶ Payload: Notify (41) - EAP_ONLY_AUTHENTICATION
      Padding (7 bytes)
      Pad Length: 7
    Integrity Checksum Data: 28b3a4126922d3201487ed10 (12 bytes) <HMAC_SHA1_96 [RFC2404]>[correct]
```

```
▼ Payload: Identification - Initiator (35)
    Next payload: Authentication (39)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 12
    ID type: IPV4_ADDR (1)
    Reserved: 000000
  ▶ Identification Data:172.16.13.1
```

It's like a tag that allows peers to match the policy and authentication data related to the other peer and fetch it from Security Policy Database (SPD) when needed.

```
▼ Payload: Authentication (39)
    Next payload: Security Association (33)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 28
    Authentication Method: Shared Key Message Integrity Code (2)
    Reserved: 000000
    Authentication Data: 6aeef8670445990e71afd5ebaa914975e8bcfe1d
```

The other side can only validate this hash if they possess the same pre-master key. The other authentication option is to use certificate or EAP.

```
Protocol (Protocol ID = ESP) -> Transform (Type: 1 ENCR -> 3DES)
                             -> Transform (Type: 3 INTEG -> AUTH_HMAC_SHA1_96)
                             -> Transform (Type: 5 ESN -> No Extended Sequence Numbers)
```

For CHILD SAs (production traffic)

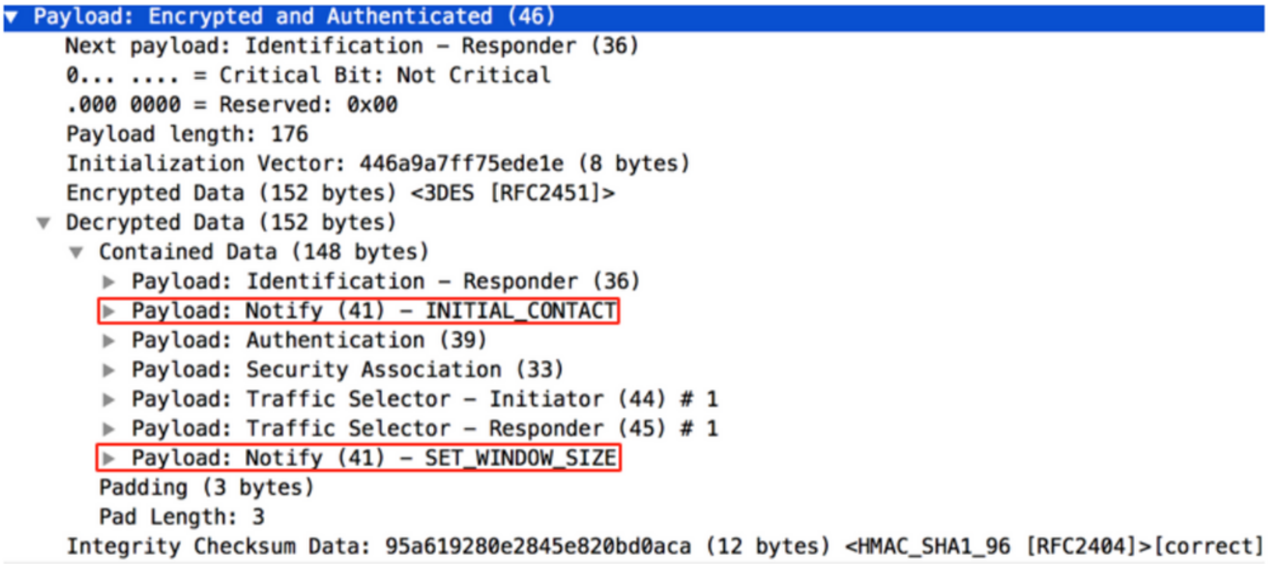
```
▼ Traffic Selector: TS_IPV4_ADDR_RANGE
    Traffic Selector Type: TS_IPV4_ADDR_RANGE (7)
    Protocol ID: Unused
    Selector Length: 16
    Start Port: 0
    End Port: 65535
    Starting Addr: 10.11.1.0
    Ending Addr: 10.11.1.255
```

IP/Port range that triggers IPSec tunnel when it belongs to Initiator.

```
▼ Traffic Selector: TS_IPV4_ADDR_RANGE
    Traffic Selector Type: TS_IPV4_ADDR_RANGE (7)
    Protocol ID: Unused
    Selector Length: 16
    Start Port: 0
    End Port: 65535
    Starting Addr: 10.13.1.0
    Ending Addr: 10.13.1.255
```

IP/Port range that triggers IPSec tunnel when it belongs to Responder.

Now, last **IKE_AUTH** message sent by Responder confirms which security parameters it picked (**Security Association** message), repeats the same **Traffic Selector** messages (if correctly configured) and sends hash of message using pre-master key (**Authentication** message)



```
▼ Payload: Encrypted and Authenticated (46)
    Next payload: Identification - Responder (36)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 176
    Initialization Vector: 446a9a7ff75ede1e (8 bytes)
    Encrypted Data (152 bytes) <3DES [RFC2451]>
  ▼ Decrypted Data (152 bytes)
    ▼ Contained Data (148 bytes)
      ▶ Payload: Identification - Responder (36)
      ▶ Payload: Notify (41) - INITIAL_CONTACT
      ▶ Payload: Authentication (39)
      ▶ Payload: Security Association (33)
      ▶ Payload: Traffic Selector - Initiator (44) # 1
      ▶ Payload: Traffic Selector - Responder (45) # 1
      ▶ Payload: Notify (41) - SET_WINDOW_SIZE
      Padding (3 bytes)
      Pad Length: 3
    Integrity Checksum Data: 95a619280e2845e820bd0aca (12 bytes) <HMAC_SHA1_96 [RFC2404]>[correct]
```

Note that I highlighted 2 Notify messages.

The **INITIAL_CONTACT** signals to Initiator that this is the only IKE_SA currently active between these peers and if there is any other IKE_SA it should be terminated in favour of this one.

The **SET_WINDOW_SIZE** is a flow control mechanism introduced in IKEv2 that allows the other side to send as many outstanding requests as the other peer wants within the window size without receiving any message acknowledging the receipt.

From now on, if additional CHILD_SAs are needed, a message called **CREATE_CHILD_SA** can be used to establish additional CHILD_SAs

It can also be used to rekey **IKE_SA** where **Notification** payload is sent of type **REKEY_SA** followed by **CREATE_CHILD_SA** with new key information so new SA is established and old one is subsequently deleted.

Security

APM BIG-IP ike ikev2 ipsec vpn wireshark

3 Kudos

Version history

Last update:

22-Jul-2019 05:08

Updated by:

Rodrigo_Albuque

Contributors

Rodrigo_Albuque

ABOUT DEVCENTRAL

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

F5 RESOURCES

Product Documentation
White Papers
Glossary
Customer Stories
Webinars
Free Online Courses
F5 Certification
LearnF5 Training

F5 SUPPORT

Manage Subscriptions
Support Portal
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