

# ASA 및 Strongswan을 사용하여 사이트 대 사이트 VPN 터널 설정

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[구성](#)

[시나리오](#)

[네트워크 다이어그램](#)

[ASA 컨피그레이션](#)

[strongSwan 컨피그레이션](#)

[유용한 명령\(strongswan\)](#)

[다음을 확인합니다.](#)

[ASA에서](#)

[1단계 확인](#)

[2단계 검증](#)

[강력한 Swan에서](#)

[문제 해결](#)

[ASA 디버그](#)

[strongSwan 디버그](#)

[관련 정보](#)

---

## 소개

이 문서에서는 ASA와 strongSwan 서버 간에 CLI를 통해 Site-to-Site IPSec Internet Key Exchange Version 1 터널을 구성하는 방법에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Cisco ASA(Adaptive Security Appliance)
- 기본 Linux 명령
- 일반적인 IPSec 개념

## 사용되는 구성 요소

이 문서의 정보는 다음 버전을 기반으로 합니다.

- 9.12(3)9를 실행하는 Cisco ASA
- Ubuntu 20.04 실행 strongSwan U5.8.2

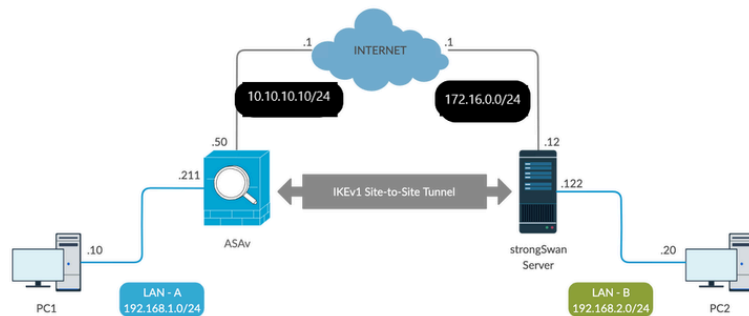
이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 구성


이 섹션에서는 ASA 및 strongSwan 컨피그레이션을 완료하는 방법에 대해 설명합니다.

### 시나리오

이 설정에서는 LAN-A의 PC1이 LAN-B의 PC2와 통신하려고 합니다. 이 트래픽은 암호화하여 ASA와 strongSwan 서버 간의 IKEv1(Internet Key Exchange Version 1) 터널을 통해 전송해야 합니다. 두 피어 모두 PSK(Pre-shared-key)로 서로를 인증합니다.



## 네트워크 다이어그램

 참고: 내부 및 외부 네트워크, 특히 사이트 간 VPN 터널을 설정하는 데 사용되는 원격 피어에 대한 연결이 있는지 확인하십시오. 기본 연결을 확인하려면 ping을 사용할 수 있습니다.

## ASA 컨피그레이션

<#root>

```
!Configure the ASA interfaces
```

```
!  
interface GigabitEthernet0/0  
nameif inside
```

```
security-level 100
ip address 192.168.1.211 255.255.255.0
!
interface GigabitEthernet0/1
nameif outside
security-level 0
ip address 10.10.10.10 255.255.255.0
!

!Configure the ACL for the VPN traffic of interest

!
object-group network local-network
network-object 192.168.1.0 255.255.255.0
!
object-group network remote-network
network-object 192.168.2.0 255.255.255.0
!
access-list asa-strongswan-vpn extended permit ip object-group local-network object-group remote-network
!

!Enable IKEv1 on the 'Outside' interface

!
crypto ikev1 enable outside
!

!Configure how ASA identifies itself to the peer

!
crypto isakmp identity address
!

!Configure the IKEv1 policy

!
crypto ikev1 policy 10
authentication pre-share
encryption aes-256
hash sha
group 5
lifetime 3600
!

!Configure the IKEv1 transform-set

!
crypto ipsec ikev1 transform-set tset esp-aes-256 esp-sha-hmac
!


!Configure a crypto map and apply it to outside interface

!
crypto map outside_map 10 match address asa-strongswan-vpn
crypto map outside_map 10 set peer 172.16.0.0
crypto map outside_map 10 set ikev1 transform-set tset
crypto map outside_map 10 set security-association lifetime seconds 28800
crypto map outside_map interface outside
!
```

```
!Configure the Tunnel group (LAN-to-LAN connection profile)
```

```
!  
tunnel-group 172.16.0.0 type ipsec-l2l  
tunnel-group 172.16.0.0 ipsec-attributes  
ikev1 pre-shared-key cisco  
!
```

---

 참고: 두 피어의 두 정책이 모두 동일한 인증, 암호화, 해시 및 Diffie-Hellman 매개변수 값을 포함할 경우 IKEv1 정책 일치가 존재합니다. IKEv1의 경우 원격 피어 정책도 개시자가 전송하는 정책의 수명보다 작거나 같은 수명을 지정해야 합니다. 수명이 동일하지 않으면 ASA에서 더 짧은 수명을 사용합니다. 또한 지정된 정책 매개변수에 대한 값을 지정하지 않으면 기본값이 적용됩니다.

---

 참고: VPN 트래픽에 대한 ACL은 NAT(Network Address Translation) 이후에 소스 및 목적지 IP 주소를 사용합니다.

---

NAT 예외(선택 사항):

일반적으로 VPN 트래픽에 대해 수행되는 NAT는 없어야 합니다. 해당 트래픽을 제외하려면 ID NAT 규칙을 생성해야 합니다. 아이덴티티 NAT 규칙은 단순히 주소를 동일한 주소로 변환합니다.

```
<#root>
```

```
nat (inside,outside) source static  
local-network local-network  
destination static  
remote-network remote-network  
no-proxy-arp route-lookup
```

## strongSwan 컨피그레이션

Ubuntu에서는 IPsec 터널에서 사용할 컨피그레이션 매개변수를 사용하여 이 두 파일을 수정합니다. 즐겨찾는 편집기를 사용하여 편집할 수 있습니다.

```
/etc/ipsec.conf
```

```
/etc/ipsec.secrets
```

```
<#root>
```

```
# /etc/ipsec.conf - strongSwan IPsec configuration file
```

# basic configuration

config setup

strictcrlpolicy=no  
uniqueids = yes  
charondebug = "all"

# VPN to ASA

conn vpn-to-asa

authby=secret  
left=%defaultroute  
leftid=172.16.0.0  
leftsubnet=192.168.2.0/24  
right=10.10.10.10  
rightid=10.10.10.10  
rightsubnet=192.168.1.0/24  
ike=aes256-sha1-modp1536  
esp=aes256-sha1  
keyingtries=%forever  
leftauth=psk  
rightauth=psk  
keyexchange=ikev1  
ikelifetime=1h  
lifetime=8h  
dpdelay=30  
dpdtimeout=120  
dpdaction=restart  
auto=start

# config setup

- Defines general configuration parameters.

# strictcrlpolicy

- Defines if a fresh CRL must be available in order for the peer authentication based on RSA signatures to succeed.

# uniqueids

- Defines whether a particular participant ID must be kept unique, with any new IKE\_SA using an ID deemed to replace all old ones using that ID.

# charondebug

- Defines how much charon debugging output must be logged.

# conn

- Defines a connection.

# authby -

Defines how the peers must authenticate; acceptable values are secret or psk, pubkey, rsasig, ecdsasig

# left -

Defines the IP address of the strongSwan's interface participating in the tunnel.

# lefid -

Defines the identity payload for the strongSwan.

# leftsubnet -

Defines the private subnet behind the strongSwan, expressed as network/netmask.

# right -

Defines the public IP address of the VPN peer.

# rightid -

Defines the identity payload for the VPN peer.

# rightsubnet -

Defines the private subnet behind the VPN peer, expressed as network/netmask.

# ike -

Defines the IKE/ISAKMP SA encryption/authentication algorithms. You can add a comma-separated list.

# esp -

Defines the ESP encryption/authentication algorithms. You can add a comma-separated list.

# keyingtries -

Defines the number of attempts that must be made to negotiate a connection.

# keyexchange -

Defines the method of key exchange, whether IKEv1 or IKEv2.

# ikelifetime -

Defines the duration of an established phase-1 connection.

# lifetime -

Defines the duration of an established phase-2 connection.

# dpddelay -

Defines the time interval with which R\_U\_THERE messages/INFORMATIONAL exchanges are sent to the peer. These are only sent if no other traffic is received.

# dpdtimeout -

Defines the timeout interval, after which all connections to a peer are deleted in case of inactivity.

# dpdaction -

Defines what action needs to be performed on DPD timeout. Takes three values as parameters :

clear

,

hold

, and  
restart.

With  
clear

the connection is closed with no further actions taken,  
hold

installs a trap policy, which catches  
matching traffic and tries to re-negotiate the connection on demand and  
restart

immediately triggers an attempt  
to re-negotiate the connection. The default is  
none

which disables the active sending of DPD messages.

# auto -

Defines what operation, if any, must be done automatically at IPsec startup (  
start

loads a connection and brings  
it up immediately).

<#root>

/etc/ipsec.secrets -

This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host which knows the public part.

172.16.0.0 10.10.10.10 : PSK "cisco"

유용한 명령(strongswan)

시작/중지/상태:

\$ sudo ipsec up <connection-name>

<#root>

```
$ sudo ipsec up vpn-to-asa
```

```
generating QUICK_MODE request 656867907 [ HASH SA No ID ID ]
sending packet: from 172.16.0.0[500] to 10.10.10.10[500] (204 bytes)
received packet: from 10.10.10.10[500] to 172.16.0.0[500] (188 bytes)
parsed QUICK_MODE response 656867907 [ HASH SA No ID ID N((24576)) ]
selected proposal: ESP:AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ
detected rekeying of CHILD_SA vpn-to-asa{2}
CHILD_SA vpn-to-asa{3} established with SPIs c9080c93_i 3f570a23_o and TS 192.168.2.0/24 === 192.168.1.0/24
connection 'vpn-to-asa' established successfully
```

```
$ sudo ipsec down <connection-name>
```

```
<#root>
```

```
$ sudo ipsec down vpn-to-asa
```

```
generating QUICK_MODE request 656867907 [ HASH SA No ID ID ]
sending packet: from 172.16.0.0[500] to 10.10.10.10[500] (204 bytes)
received packet: from 10.10.10.10[500] to 172.16.0.0[500] (188 bytes)
parsed QUICK_MODE response 656867907 [ HASH SA No ID ID N((24576)) ]
selected proposal: ESP:AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ
detected rekeying of CHILD_SA vpn-to-asa{2}
CHILD_SA vpn-to-asa{3} established with SPIs c9080c93_i 3f570a23_o and TS 192.168.2.0/24 === 192.168.1.0/24
connection 'vpn-to-asa' established successfully
anurag@strongswan214:~$ sudo ipsec down vpn-to-asa
closing CHILD_SA vpn-to-asa{3} with SPIs c9080c93_i (0 bytes) 3f570a23_o (0 bytes) and TS 192.168.2.0/24 === 192.168.1.0/24
sending DELETE for ESP CHILD_SA with SPI c9080c93
generating INFORMATIONAL_V1 request 3465984663 [ HASH D ]
sending packet: from 172.16.0.0[500] to 10.10.10.10[500] (76 bytes)
deleting IKE_SA vpn-to-asa[2] between 172.16.0.0[172.16.0.0]...10.10.10.10[10.10.10.10]
sending DELETE for IKE_SA vpn-to-asa[2]
generating INFORMATIONAL_V1 request 2614622058 [ HASH D ]
sending packet: from 172.16.0.0[500] to 10.10.10.10[500] (92 bytes)
IKE_SA [2] closed successfully
```

```
$ sudo ipsec 다시 시작
```

```
Stopping strongSwan IPsec...
Starting strongSwan 5.8.2 IPsec [starter]...
```

```
$ sudo ipsec 상태
```



Security Associations (1 up, 0 connecting):

```
vpn-to-asa[1]: ESTABLISHED 35 seconds ago, 172.16.0.0[172.16.0.0]...10.10.10.10[10.10.10.10]
vpn-to-asa{1}: REKEYED, TUNNEL, reqid 1, expires in 7 hours
vpn-to-asa{1}: 192.168.2.0/24 === 192.168.1.0/24
vpn-to-asa{2}: INSTALLED, TUNNEL, reqid 1, ESP SPIs: c0d93265_i 599b4d60_o
vpn-to-asa{2}: 192.168.2.0/24 === 192.168.1.0/24
```

\$ sudo ipsec 상태모두

Status of IKE charon daemon (strongSwan 5.8.2, Linux 5.4.0-37-generic, x86\_64):

```
uptime: 2 minutes, since Jun 27 07:15:14 2020
malloc: sbrk 2703360, mmap 0, used 694432, free 2008928
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 3
loaded plugins: charon aesni aes rc2 sha2 sha1 md5 mgf1 random nonce x509 revocation constraints pubkey
Listening IP addresses:
```

```
172.16.0.0
192.168.2.122
```

Connections:

```
vpn-to-asa: %any...10.10.10.10 IKEv1, dpddelay=30s
vpn-to-asa: local: [172.16.0.0] uses pre-shared key authentication
vpn-to-asa: remote: [10.10.10.10] uses pre-shared key authentication
vpn-to-asa: child: 192.168.2.0/24 === 192.168.1.0/24 TUNNEL, dpdaction=restart
```

Security Associations (1 up, 0 connecting):

```
vpn-to-asa[1]: ESTABLISHED 2 minutes ago, 172.16.0.0[172.16.0.0]...10.10.10.10[10.10.10.10]
vpn-to-asa[1]: IKEv1 SPIs: 57e24d839bf05f95_i* 6a4824492f289747_r, pre-shared key reauthentication in 4
vpn-to-asa[1]: IKE proposal: AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1536
vpn-to-asa{2}: INSTALLED, TUNNEL, reqid 1, ESP SPIs: c0d93265_i 599b4d60_o
vpn-to-asa{2}: AES_CBC_256/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 7 hours
vpn-to-asa{2}: 192.168.2.0/24 === 192.168.1.0/24
```

IPsec 터널의 정책 및 상태를 가져옵니다.

\$ sudo ip xfrm 상태

```
src 172.16.0.0 dst 10.10.10.10
proto esp spi 0x599b4d60 reqid 1 mode tunnel
replay-window 0 flag af-unspec
auth-trunc hmac(sha1) 0x52c84359280868491a37e966384e4c6db05384c8 96
enc cbc(aes) 0x99e00f0989fec6baa7bd4ea1c7fbefdf37f04153e721a060568629e603e23e7a
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
src 10.10.10.10 dst 172.16.0.0
proto esp spi 0xc0d93265 reqid 1 mode tunnel
replay-window 32 flag af-unspec
auth-trunc hmac(sha1) 0x374d9654436a4c4fe973a54da044d8814184861e 96
enc cbc(aes) 0xf51a4887281551a246a73c3518d938fd4918928088a54e2abc5253bd2de30fd6
anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
```

\$ sudo ip xfrm 정책

```
src 192.168.2.0/24 dst 192.168.1.0/24
dir out priority 375423
tmpl src 172.16.0.0 dst 10.10.10.10
proto esp spi 0x599b4d60 reqid 1 mode tunnel
src 192.168.1.0/24 dst 192.168.2.0/24
dir fwd priority 375423
tmpl src 10.10.10.10 dst 172.16.0.0
proto esp reqid 1 mode tunnel
src 192.168.1.0/24 dst 192.168.2.0/24
dir in priority 375423
tmpl src 10.10.10.10 dst 172.16.0.0
proto esp reqid 1 mode tunnel
src 0.0.0.0/0 dst 0.0.0.0/0
socket in priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket out priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket in priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
socket out priority 0
src ::/0 dst ::/0
socket in priority 0
src ::/0 dst ::/0
socket out priority 0
src ::/0 dst ::/0
socket in priority 0
src ::/0 dst ::/0
socket out priority 0
```

서비스가 실행되는 동안 암호를 다시 로드합니다.

\$ sudo ipsec readsecrets

트래픽이 터널을 통과하는지 확인합니다.


\$ sudo tcpdump esp

```
09:30:27.788533 IP 172.16.0.0 > 10.10.10.10: ESP(spi=0x599b4d60,seq=0x1e45), length 132
09:30:27.788779 IP 172.16.0.0 > 10.10.10.10: ESP(spi=0x599b4d60,seq=0x1e45), length 132
09:30:27.790348 IP 10.10.10.10 > 172.16.0.0: ESP(spi=0xc0d93265,seq=0x11), length 132
09:30:27.790512 IP 10.10.10.10 > 172.16.0.0: ESP(spi=0xc0d93265,seq=0x11), length 132
09:30:28.788946 IP 172.16.0.0 > 10.10.10.10: ESP(spi=0x599b4d60,seq=0x1e46), length 132
09:30:28.789201 IP 172.16.0.0 > 10.10.10.10: ESP(spi=0x599b4d60,seq=0x1e46), length 132
09:30:28.790116 IP 10.10.10.10 > 172.16.0.0: ESP(spi=0xc0d93265,seq=0x12), length 132
09:30:28.790328 IP 10.10.10.10 > 172.16.0.0: ESP(spi=0xc0d93265,seq=0x12), length 132
```

## 다음을 확인합니다.

터널이 작동 중인지, 트래픽을 전달하는지 확인하기 전에 해당 트래픽이 ASA 또는 strongSwan 서버로 전송되는지 확인해야 합니다.

---

 참고: ASA에서는 IPSec 터널을 시작하기 위해 관심 트래픽과 일치하는 패킷 추적기 도구를 사용할 수 있습니다(예: tcp 192.168.1.100 12345 192.168.2.200 80 내부 패킷 추적기 입력).

---

## ASA에서

### 1단계 확인

ASA에서 IKEv1 1단계가 작동 중인지 확인하려면 `how crypto ikev1 sa`(또는 `show crypto isakmp sa`) 명령을 입력합니다. `MM_ACTIVEstate`가 표시되어야 합니다.

<#root>

ASAv#

```
show crypto ikev1 sa
```

IKEv1 SAs:

Active SA: 1

Rekey SA: 0 (A tunnel will report 1 Active and 1 Rekey SA during rekey)

Total IKE SA: 1

1 IKE Peer:

172.16.0.0

Type : L2L Role : responder


Rekey : no State :

**MM\_ACTIVE**

### 2단계 검증

ASA에서 IKEv1 2단계가 작동 중인지 확인하려면 암호화 `ipsec sa` 표시 명령을 실행합니다. 필요한 출력은 인바운드 및 아웃바운드 SPI(Security Parameter Index)를 모두 보는 것입니다. 트래픽이 터널을 통과하는 경우 `encaps/decaps` 카운터가 증가해야 합니다.

---

 참고: 각 ACL 항목에 대해 별도의 인바운드/아웃바운드 SA가 생성되며, 이는 긴 `show crypto ipsec sa` 명령 출력을 초래할 수 있습니다(암호화 ACL의 ACE 항목 수에 따라 다름).

---

<#root>

ASAv#

show crypto ipsec sa peer 172.16.0.0

interface:

outside

Crypto map tag: outside\_map, seq num: 10, local addr: 10.10.10.10

access-list asa-strongswan-vpn extended permit ip 192.168.1.0 255.255.255.0 192.168.2.0 255.255.255.0  
local ident (addr/mask/prot/port): (

192.168.1.0

/255.255.255.0/0/0)

remote ident (addr/mask/prot/port): (

192.168.2.0

/255.255.255.0/0/0)

current\_peer:

172.16.0.0

#

pkts encaps: 37, #pkts encrypt: 37, #pkts digest: 37

#

pkts decaps: 37, #pkts decrypt: 37, #pkts verify: 37

#pkts compressed: 0, #pkts decompressed: 0

#pkts not compressed: 37, #pkts comp failed: 0, #pkts decomp failed: 0

#pre-frag successes: 0, #pre-frag failures: 0, #fragments created: 0

#PMTUs sent: 0, #PMTUs rcvd: 0, #decapsulated frgs needing reassembly: 0

#TFC rcvd: 0, #TFC sent: 0

#Valid ICMP Errors rcvd: 0, #Invalid ICMP Errors rcvd: 0

#send errors: 0, #recv errors: 0

local crypto endpt.: 10.10.10.10/0, remote crypto endpt.:

172.16.0.0

/0

path mtu 1500, ipsec overhead 74(44), media mtu 1500

PMTU time remaining (sec): 0, DF policy: copy-df

ICMP error validation: disabled, TFC packets: disabled

current outbound spi: C8F1BFAB

current inbound spi : 3D64961A

inbound esp sas:

```
spi: 0x3D64961A (1030002202)
SA State: active
transform: esp-aes-256 esp-sha-hmac no compression
in use settings ={L2L, Tunnel, IKEv1, }
slot: 0, conn_id: 31, crypto-map: outside_map
sa timing: remaining key lifetime (kB/sec): (4373997/27316)
IV size: 16 bytes
replay detection support: Y
Anti replay bitmap:
0x000001FF 0xFFFFFFFF
outbound esp sas:
spi: 0xC8F1BFAB (3371286443)
SA State: active
transform: esp-aes-256 esp-sha-hmac no compression
in use settings ={L2L, Tunnel, IKEv1, }
slot: 0, conn_id: 31, crypto-map: outside_map
sa timing: remaining key lifetime (kB/sec): (4373997/27316)
IV size: 16 bytes
replay detection support: Y
Anti replay bitmap:
0x00000000 0x00000001
```

또는 show vpn-sessiondb 명령을 사용하여 1단계와 2단계 모두의 세부사항을 함께 확인할 수 있습니다.

```
<#root>
```

```
ASAv#
```

```
show vpn-sessiondb detail l2l filter ipaddress 172.16.0.0
```

```
Session Type: LAN-to-LAN Detailed
```

```
Connection :
```

```
172.16.0.0
```

```
Index : 3 IP Addr : 172.16.0.0
```

```
Protocol :
```

```
IKEv1 IPsec
```

```
Encryption : IKEv1: (1)AES256 IPsec: (1)AES256
```

```
Hashing : IKEv1: (1)SHA1 IPsec: (1)SHA1
```

```
Bytes Tx : 536548 Bytes Rx : 536592
```

```
Login Time : 12:45:14 IST Sat Jun 27 2020
```

```
Duration : 1h:51m:57s
```

```
IKEv1 Tunnels: 1
```

```
IPsec Tunnels: 1
```

```
IKEv1:
```

```
Tunnel ID : 3.1
```

UDP Src Port : 500 UDP Dst Port : 500

IKE Neg Mode : Main Auth Mode : preSharedKeys

Encryption : AES256 Hashing : SHA1  
Rekey Int (T): 3600 Seconds Rekey Left(T): 2172 Seconds  
D/H Group : 5  
Filter Name :

IPsec:  
Tunnel ID : 3.2

Local Addr : 192.168.1.0/255.255.255.0/0/0

Remote Addr : 192.168.2.0/255.255.255.0/0/0

Encryption : AES256 Hashing : SHA1  
Encapsulation: Tunnel  
Rekey Int (T): 28800 Seconds Rekey Left(T): 22099 Seconds  
Rekey Int (D): 4608000 K-Bytes Rekey Left(D): 4607476 K-Bytes  
Idle Time Out: 30 Minutes Idle TO Left : 30 Minutes  
Bytes Tx : 536638 Bytes Rx : 536676  
Pkts Tx : 6356 Pkts Rx : 6389

## 강력한 Swan에서

```
<#root>
```

```
#
```

```
sudo ipsec statusall
```

```
Status of IKE charon daemon (strongSwan 5.8.2, Linux 5.4.0-37-generic, x86_64):  
uptime: 2 minutes, since Jun 27 07:15:14 2020  
malloc: sbrk 2703360, mmap 0, used 694432, free 2008928  
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 3  
loaded plugins: charon aesni aes rc2 sha2 sha1 md5 mgf1 random nonce x509 revocation constraints pubkey  
Listening IP addresses:  
172.16.0.0  
192.168.2.122  
Connections:  
vpn-to-asa: %any...10.10.10.10 IKEv1, dpddelay=30s  
vpn-to-asa:  
  
local: [172.16.0.0]  
  
uses pre-shared key authentication  
vpn-to-asa:  
  
remote: [10.10.10.10]  
  
uses pre-shared key authentication  
vpn-to-asa:  
  
child: 192.168.2.0/24 === 192.168.1.0/24 TUNNEL
```

```
, dpdaction=restart
Security Associations (1 up, 0 connecting):
vpn-to-asa[1]:

ESTABLISHED

 2 minutes ago, 172.16.0.0[172.16.0.0]...10.10.10.10[10.10.10.10]
vpn-to-asa[1]: IKEv1 SPIs: 57e24d839bf05f95_i* 6a4824492f289747_r, pre-shared key reauthentication in 4
vpn-to-asa[1]: IKE proposal: AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1536
vpn-to-asa{2}:

INSTALLED, TUNNEL,

 reqid 1, ESP SPIs: c0d93265_i 599b4d60_o
vpn-to-asa{2}: AES_CBC_256/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 7 hours
vpn-to-asa{2}:


192.168.2.0/24 === 192.168.1.0/24
```

## 문제 해결

### ASA 디버그

ASA 방화벽에서 IPsec IKEv1 터널 협상 문제를 해결하려면 다음 명령을 사용할 수 있습니다.

---


 주의: ASA에서는 다양한 디버그 레벨을 설정할 수 있습니다. 기본적으로 레벨 1이 사용됩니다. 디버그 수준을 변경하면 디버그의 세부 정도가 증가할 수 있습니다. 이 경우 레벨 127은 트러블슈팅을 위한 충분한 세부사항을 제공합니다. 특히 프로덕션 환경에서는 이 작업을 신중하게 수행해야 합니다.

---

```
<#root>
```

```
debug crypto ipsec 127
debug crypto isakmp 127
debug ike-common 10
```

---

 참고: ASA에 여러 VPN 터널이 있는 경우 디버그 출력을 지정된 피어만 포함하도록 제한하려면 조건부 디버그(디버그 암호화 조건 피어 A.B.C.D)를 사용하는 것이 좋습니다.

---

### strongSwan 디버그

다음과 같이 charon debug가 ipsec.conf 파일에서 활성화되었는지 확인합니다.

```
<#root>
```

```
charondebug = "all"
```

로그 메시지가 결국 끝나는 위치는 시스템에서 syslog를 구성하는 방법에 따라 달라집니다. 일반적인 위치는 /var/log/daemon, /var/log/syslog 또는 /var/log/messages입니다.

## 관련 정보

- [strongSwan 사용자 설명서](#)
- [Cisco IOS®와 strongSwan 간의 IKEv1/IKEv2 컨피그레이션 예](#)
- [ASA와 Cisco IOS® 라우터 간에 Site-to-Site IPSec IKEv1 터널 구성](#)



이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.