Konfigurieren eines standortübergreifenden VPN-Tunnels mit ASA und Strongswan

Inhalt

Einleitung

In diesem Dokument wird beschrieben, wie ein Site-to-Site IPSec Internet Key Exchange Version 1-Tunnel über die CLI zwischen einem ASA- und einem strongSwan-Server konfiguriert wird.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Cisco Adaptive Security Appliance (ASA)
- Grundlegende Linux-Befehle
- Allgemeine IPSec-Konzepte

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Versionen:

- Cisco ASAv mit 9.12(3)9
- Ubuntu 20.04 mit strongSwan U5.8.2

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle verstehen.

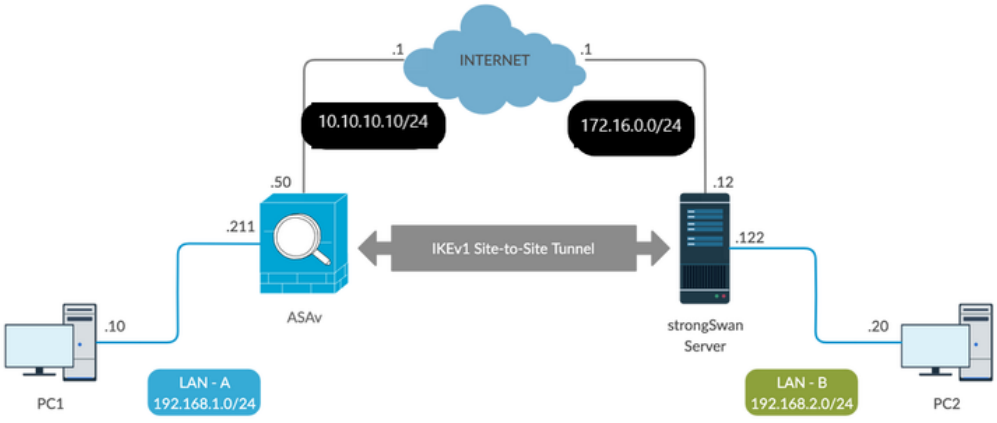
Konfigurieren

In diesem Abschnitt wird beschrieben, wie Sie die ASA- und strongSwan-Konfigurationen abschließen.

Szenario

In dieser Konfiguration möchte PC1 in LAN-A mit PC2 in LAN-B kommunizieren. Dieser Datenverkehr muss verschlüsselt und über einen IKEv1-Tunnel (Internet Key Exchange Version 1) zwischen ASA und StongSwan-Server gesendet werden. Beide Peers authentifizieren sich gegenseitig mit einem Pre-Shared Key (PSK).

Netzwerkdiagramm



> **Anmerkung:** Stellen Sie sicher, dass eine Verbindung sowohl zu internen als auch zu externen Netzwerken besteht, insbesondere zum Remote-Peer, der für die Einrichtung eines Site-to-Site-VPN-Tunnels verwendet wird. Sie können einen Ping verwenden, um die grundlegenden Netzwerkverbindungen zu überprüfen.

ASA-Konfiguration

```
!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
!
```

Anmerkung: Eine IKEv1-Richtlinienübereinstimmung liegt vor, wenn beide Richtlinien der beiden Peers dieselben Werte für Authentifizierung, Verschlüsselung, Hash und Diffie-Hellman-Parameter enthalten. Für IKEv1 muss die Remote-Peer-Richtlinie auch eine Lebensdauer angeben, die kleiner oder gleich der Lebensdauer in der vom Initiator gesendeten Richtlinie ist. Sind die Lebensdauern nicht identisch, verwendet die ASA eine kürzere Lebensdauer. Wenn Sie für einen bestimmten Richtlinienparameter keinen Wert angeben, wird der Standardwert angewendet.

Hinweis: Eine ACL für VPN-Datenverkehr verwendet die Quell- und Ziel-IP-Adressen nach Network Address Translation (NAT).

NAT-Ausnahme (optional):

In der Regel darf für den VPN-Verkehr keine NAT durchgeführt werden. Sie müssen eine Identitäts-NAT-Regel erstellen, um diesen Datenverkehr auszuschließen. Die Identitäts-NAT-Regel übersetzt einfach eine Adresse in dieselbe Adresse.

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

strongSwan-Konfiguration

Unter Ubuntu würden Sie diese beiden Dateien mit Konfigurationsparametern ändern, die im IPsec-Tunnel verwendet werden. Sie können Ihre Favoriten mit Ihrem Editor bearbeiten.

/etc/ipsec.conf

/etc/ipsec.secrets

```
# /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
             dpddelay=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 paricipating 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 paramters : **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).

/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"

Nützliche Befehle (strongswan)

Start/Stopp/Status:

$ sudo ipsec up <Verbindungsname>

$ 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 <Verbindungsname>

```
$ 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-Neustart

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

$ sudo ipsec-Status

```
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 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 pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf gmp agent xcbc hmac gcm
drbg attr kernel-netlink resolve socket-default connmark stroke updown eap-mschapv2 xauth-
generic counters
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 40 minutes
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
```

Abrufen der Richtlinien und des Status des IPsec-Tunnels:

$ sudo ip xfrm status

```
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 richtlinie

```
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
```

Laden Sie die Geheimnisse neu, während der Dienst ausgeführt wird:

$ sudo ipsec readsecrets

Überprüfen Sie, ob der Datenverkehr durch den Tunnel fließt:

$ 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
```

Überprüfung

Bevor Sie überprüfen, ob der Tunnel in Betrieb ist und den Datenverkehr weiterleitet, müssen Sie sicherstellen, dass der relevante Datenverkehr entweder an die ASA oder den strongSwan-Server gesendet wird.

> **Hinweis:** Auf der ASA kann das Tool PacketTracer, das dem relevanten Datenverkehr entspricht, verwendet werden, um den IPSec-Tunnel zu initiieren (z. B. **packet-tracer input inside tcp tcp 192.168.1.100 12345 192.168.2.200 80 detailed).**

Auf ASA

Überprüfung Phase 1

Um zu überprüfen, ob IKEv1 Phase 1 auf der ASA aktiv ist, geben Sie den Befehl **show crypto ikev1 sa (**oder **show crypto isakmp sa)** ein. Für die erwartete Ausgabe wird **derMM_**ACTIVEstate angezeigt:

```
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
```

Überprüfung Phase 2

Um zu überprüfen, ob IKEv1 Phase 2 auf der ASA aktiv ist, geben Sie den **show crypto ipsec sa** aus. Als Ausgabe wird sowohl der eingehende als auch der ausgehende Security Parameter Index (SPI) erwartet. Wenn der Datenverkehr den Tunnel durchläuft, müssen die Zähler für Encaps/Entcaps inkrementiert werden.

> **Hinweis**: Für jeden ACL-Eintrag wird eine separate eingehende/ausgehende SA erstellt, was zu einer langen Ausgabe des Befehls **show crypto ipsec sa** führen kann (abhängig von der Anzahl der ACE-Einträge in der Krypto-ACL).

```
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
```

Alternativ können Sie den Befehl **show vpn-sessiondb** verwenden, um die Details für Phase 1 und 2 gemeinsam zu überprüfen.

```
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
```

Auf starken Schwan

```
# 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 pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf gmp agent xcbc hmac gcm
drbg attr kernel-netlink resolve socket-default connmark stroke updown eap-mschapv2 xauth-
generic counters
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 40 minutes
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
```

Fehlerbehebung

ASA-Fehlersuche

Um die IPSec IKEv1-Tunnelaushandlung auf einer ASA-Firewall zu beheben, können Sie die folgendenDebug-Befehle verwenden:

> **Vorsicht:** Auf der ASA können Sie verschiedene Debug-Ebenen festlegen. Standardmäßig wird Ebene 1 verwendet. Wenn Sie die Debugstufe ändern, kann die Ausführlichkeit der Debugging-Vorgänge zunehmen. In diesem Fall bietet die Stufe 127 ausreichende Details zur Fehlerbehebung. Gehen Sie dabei besonders in Produktionsumgebungen mit Vorsicht vor.

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

> **Hinweis**: Wenn auf dem ASA mehrere VPN-Tunnel vorhanden sind, wird empfohlen, konditionales Debuggen (**debug crypto condition peer A.B.C.D)** zu verwenden, um die Debugausgaben auf den angegebenen Peer zu beschränken.

strongSwan-Debugger

Stellen Sie sicher, dass das Zeichnungsdebuggen in der Datei ipsec.conf aktiviert ist:

```
charondebug = "all"
```

Wo die Protokollmeldungen letztlich landen, hängt davon ab, wie syslog auf Ihrem System konfiguriert ist. Häufig verwendete Orte sind **/var/log/daemon**, **/var/log/syslog** oder **/var/log/messages**.

Zugehörige Informationen

- strongSwan-Benutzerdokumentation
- IKEv1/IKEv2 Zwischen Cisco IOS und strongSwan - Konfigurationsbeispiel
- Konfigurieren eines Site-to-Site IPSec-IKEv1-Tunnels zwischen einem ASA- und einem

Cisco IOS-Router

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.