IKEv2 d'Android strongSwan vers Cisco IOS avec authentification EAP et RSA

Contenu

Introduction
Conditions préalables
Conditions requises
Components Used
Configuration
Diagramme du réseau
Inscription de certificat
Logiciel Cisco IOS
Android
Authentification EAP
Configuration du logiciel Cisco IOS pour l'authentification EAP
Configuration Android pour l'authentification EAP
Test d'authentification EAP
Authentification RSA
Configuration du logiciel Cisco IOS pour l'authentification RSA
Configuration Android pour authentification RSA
Test d'authentification RSA
Passerelle VPN derrière la NAT - fortes limitations du logiciel Cisco IOS et de strongSwan
Vérification
Dépannage
strongSwan CA Multiple CERT_REQ
Source du tunnel sur DVTI
Demandes d'amélioration et de bogues du logiciel Cisco IOS
Informations connexes

Introduction

Ce document décrit comment configurer la version mobile de strongSwan afin d'accéder à une passerelle VPN logicielle Cisco IOS® via le protocole Internet Key Exchange Version 2 (IKEv2).

Trois exemples sont présentés :

- Téléphone Android avec strongSwan qui se connecte à la passerelle VPN du logiciel Cisco IOS avec authentification EAP-MD5 (Extensible Authentication Protocol).
- Téléphone Android avec strongSwan qui se connecte à la passerelle VPN du logiciel Cisco IOS avec authentification par certificat (RSA).

- Téléphone Android avec strongSwan qui se connecte à la passerelle VPN du logiciel Cisco IOS derrière la traduction d'adresses de réseau (NAT). Il est obligatoire d'avoir deux extensions x509 Subject Alternative Name dans le certificat de passerelle VPN.

La plate-forme logicielle Cisco IOS et les limites de strongSwan sont également incluses.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Connaissance de base de la configuration OpenSSL
- Connaissance de base de la configuration de l'interface de ligne de commande (CLI) du logiciel Cisco IOS
- Connaissances de base sur IKEv2

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

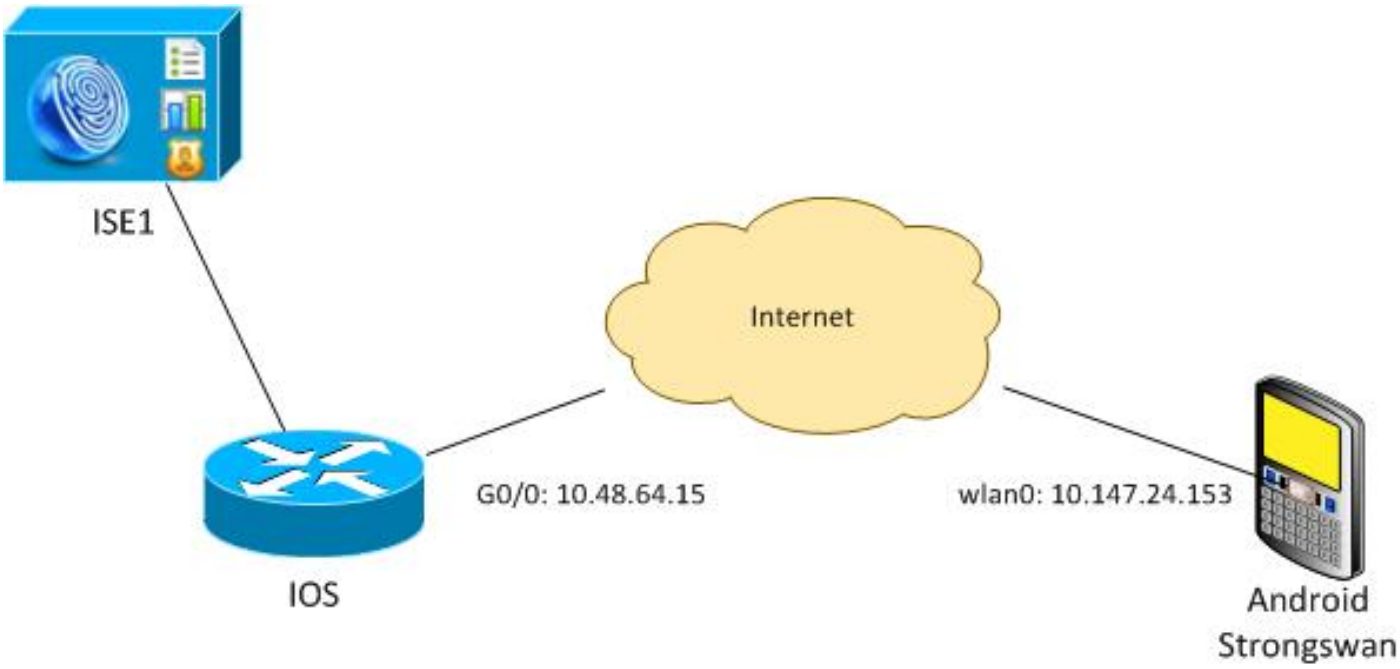
- Android 4.0 ou ultérieur avec strongSwan
- Logiciel Cisco IOS Version 15.3T ou ultérieure
- Logiciel Cisco Identity Services Engine (ISE), versions 1.1.4 et ultérieures

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Configuration

> **Remarques :**
>
> L'Outil d'interprétation de sortie (clients enregistrés seulement) prend en charge certaines commandes d'affichage. Utilisez l'Outil d'interprétation de sortie afin de visualiser une analyse de commande d'affichage de sortie .
>
> Référez-vous aux informations importantes sur les commandes de débogage avant d'utiliser les commandes de débogage.

Diagramme du réseau



Android strongSwan établit un tunnel IKEv2 avec une passerelle logicielle Cisco IOS afin d'accéder aux réseaux internes en toute sécurité.

Inscription de certificat

Les certificats sont une condition préalable à l'authentification basée sur EAP et RSA.

Dans le scénario d'authentification EAP, un certificat est nécessaire uniquement sur la passerelle VPN. Le client se connecte au logiciel Cisco IOS uniquement lorsque le logiciel présente un certificat signé par une autorité de certification qui est fiable sur Android. Une session EAP démarre ensuite pour que le client s'authentifie auprès du logiciel Cisco IOS.

Pour l'authentification basée sur RSA, les deux points de terminaison doivent avoir un certificat correct.

Lorsqu'une adresse IP est utilisée comme ID d'homologue, il existe des exigences supplémentaires pour le certificat. Android strongSwan vérifie si l'adresse IP de la passerelle VPN est incluse dans l'extension x509 Subject Alternative Name. Si ce n'est pas le cas, Android abandonne la connexion ; il s'agit d'une bonne pratique ainsi que d'une recommandation de la RFC 6125.

OpenSSL est utilisé comme autorité de certification car le logiciel Cisco IOS a une limite : il ne peut pas générer de certificats avec une extension qui inclut une adresse IP. Tous les certificats sont générés par OpenSSL et importés dans Android et le logiciel Cisco IOS.

Dans le logiciel Cisco IOS, la commande **subject-alt-name** peut être utilisée afin de créer une extension qui inclut une adresse IP, mais la commande fonctionne uniquement avec des certificats auto-signés. L'ID de bogue Cisco CSCui44783, « IOS ENH PKI ability to Generation CSR with subject-alt-name extension, » est une demande d'amélioration pour permettre au logiciel Cisco IOS de générer l'extension pour tous les types d'inscriptions.

Voici un exemple des commandes qui génèrent une autorité de certification :

```
#generate key
openssl genrsa -des3 -out ca.key 2048

#generate CSR
openssl req -new -key ca.key -out ca.csr

#remove protection
cp ca.key ca.key.org
openssl rsa -in ca.key.org -out ca.key

#self sign certificate
openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
-extensions v3_req -extfile conf_global.crt
```

conf_global.crt est un fichier de configuration. L'extension CA doit être définie sur TRUE :

```
[ req ]
default_bits              = 1024            # Size of keys
default_md                = md5                     # message digest algorithm
string_mask               = nombstr         # permitted characters
#string_mask              = pkix     # permitted characters
distinguished_name        = req_distinguished_name
req_extensions            = v3_req

[ v3_req ]
basicConstraints          = CA:TRUE
subjectKeyIdentifier      = hash
```

Les commandes qui génèrent un certificat sont très similaires pour le logiciel Cisco IOS et Android. Cet exemple suppose qu'il existe déjà une autorité de certification utilisée pour signer le certificat :

```
#generate key
openssl genrsa -des3 -out server.key 2048

#generate CSR
openssl req -new -key server.key -out server.csr

#remove protection
cp server.key server.key.org
openssl rsa -in server.key.org -out server.key

#sign the cert and add Alternate Subject Name extension from
conf_global_cert.crt file with configuration
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial
-out server.crt -days 365 -extensions v3_req -extfile conf_global_cert.crt

#create pfx file containig CA cert and server cert
openssl pkcs12 -export -out server.pfx -inkey server.key -in server.crt
-certfile ca.crt
```

conf_global_cert.crt est un fichier de configuration. L'extension Autre nom de sujet est un paramètre clé. Dans cet exemple, l'extension CA est définie sur FALSE :

```
[ req ]
default_bits              = 1024            # Size of keys
default_md                = md5                     # message digest algorithm
string_mask               = nombstr         # permitted characters
#string_mask              = pkix     # permitted characters
distinguished_name        = req_distinguished_name
```

```
req_extensions            = v3_req

[ v3_req ]
basicConstraints          = CA:FALSE
subjectKeyIdentifier      = hash
subjectAltName            = @alt_names

[alt_names]
IP.1                      = 10.48.64.15
```

Un certificat doit être généré pour le logiciel Cisco IOS et Android.

L'adresse IP 10.48.64.15 appartient à la passerelle du logiciel Cisco IOS. Lorsque vous générez un certificat pour le logiciel Cisco IOS, assurez-vous que le subjectAltName est défini sur 10.48.64.15. Android valide le certificat reçu du logiciel Cisco IOS et tente de trouver son adresse IP dans le subjectAltName.

Logiciel Cisco IOS

Le logiciel Cisco IOS doit avoir un certificat correct installé pour l'authentification basée sur RSA et EAP.

Le fichier pfx (qui est un conteneur pkcs12) du logiciel Cisco IOS peut être importé :

```
BSAN-2900-1(config)# crypto pki import TP pkcs12
http://10.10.10.1/server.pfx password 123456
% Importing pkcs12...
Source filename [server.pfx]?
CRYPTO_PKI: Imported PKCS12 file successfully.
```

Utilisez la commande **show crypto pki certificate verbose** afin de vérifier que l'importation a réussi :

```
BSAN-2900-1# show crypto pki certificates verbose
Certificate
 Status: Available
 Version: 3
 Certificate Serial Number (hex): 00A003C5DCDEFA146C
 Certificate Usage: General Purpose
 Issuer:
   cn=Cisco
   ou=Cisco TAC
   o=Cisco
   l=Krakow
   st=Malopolskie
   c=PL
  Subject:
    Name: IOS
    IP Address: 10.48.64.15
    cn=IOS
    ou=TAC
    o=Cisco
    l=Krakow
    st=Malopolska
    c=PL
 Validity Date:
   start date: 18:04:09 UTC Aug 1 2013
   end   date: 18:04:09 UTC Aug 1 2014
 Subject Key Info:
```

```
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
  Signature Algorithm: SHA1 with RSA Encryption
  Fingerprint MD5: 2C45BF10 0BACB98D 444F5804 1DC27ECF
  Fingerprint SHA1: 26B66A66 DF5E7D6F 498DD653 A2C164D7 4C7A7F8F
  X509v3 extensions:
    X509v3 Subject Key ID: AD598A9B 8AB6893B AB3CB8B9 28B2039C 78441E72
    X509v3 Basic Constraints:
        CA: FALSE
     X509v3 Subject Alternative Name:

         10.48.64.15
    Authority Info Access:
  Associated Trustpoints: TP
  Storage: nvram:Cisco#146C.cer
  Key Label: TP
  Key storage device: private config

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 00DC8EAD98723DF56A
  Certificate Usage: General Purpose
  Issuer:
    cn=Cisco
    ou=Cisco TAC
    o=Cisco
    l=Krakow
    st=Malopolskie
    c=PL
  Subject:
    cn=Cisco
    ou=Cisco TAC
    o=Cisco
    l=Krakow
    st=Malopolskie
    c=PL
  Validity Date:
    start date: 16:39:55 UTC Jul 23 2013
    end   date: 16:39:55 UTC Jul 23 2014
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
  Signature Algorithm: SHA1 with RSA Encryption
  Fingerprint MD5: 0A2432DC 33F0DC46 AAB23E26 ED474B7E
  Fingerprint SHA1: A50E3892 ED5C4542 FA7FF584 DE07B6E0 654A62D0
  X509v3 extensions:
    X509v3 Subject Key ID: 786F263C 0F5A1963 D6AD18F8 86DCE7C9 0185911E
    X509v3 Basic Constraints:
        CA: TRUE
    Authority Info Access:
  Associated Trustpoints: TP
  Storage: nvram:Cisco#F56ACA.cer

BSAN-2900-1#show ip int brief
Interface                  IP-Address      OK? Method Status          Protocol
GigabitEthernet0/0         10.48.64.15     YES NVRAM  up              up
```

Android

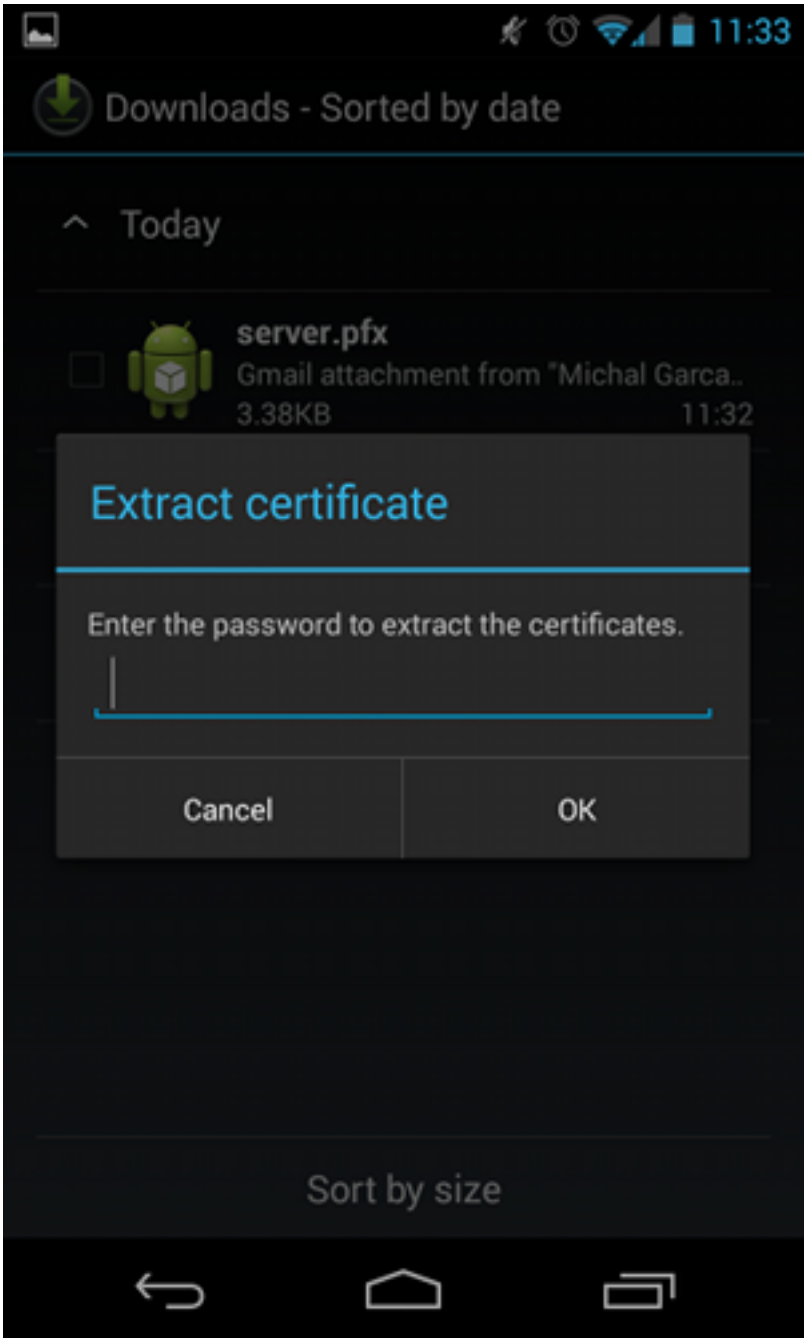
Pour l'authentification basée sur EAP, Andorid doit avoir installé le certificat CA correct.

Pour l'authentification basée sur RSA, Andorid doit avoir installé à la fois le certificat CA et son propre certificat.

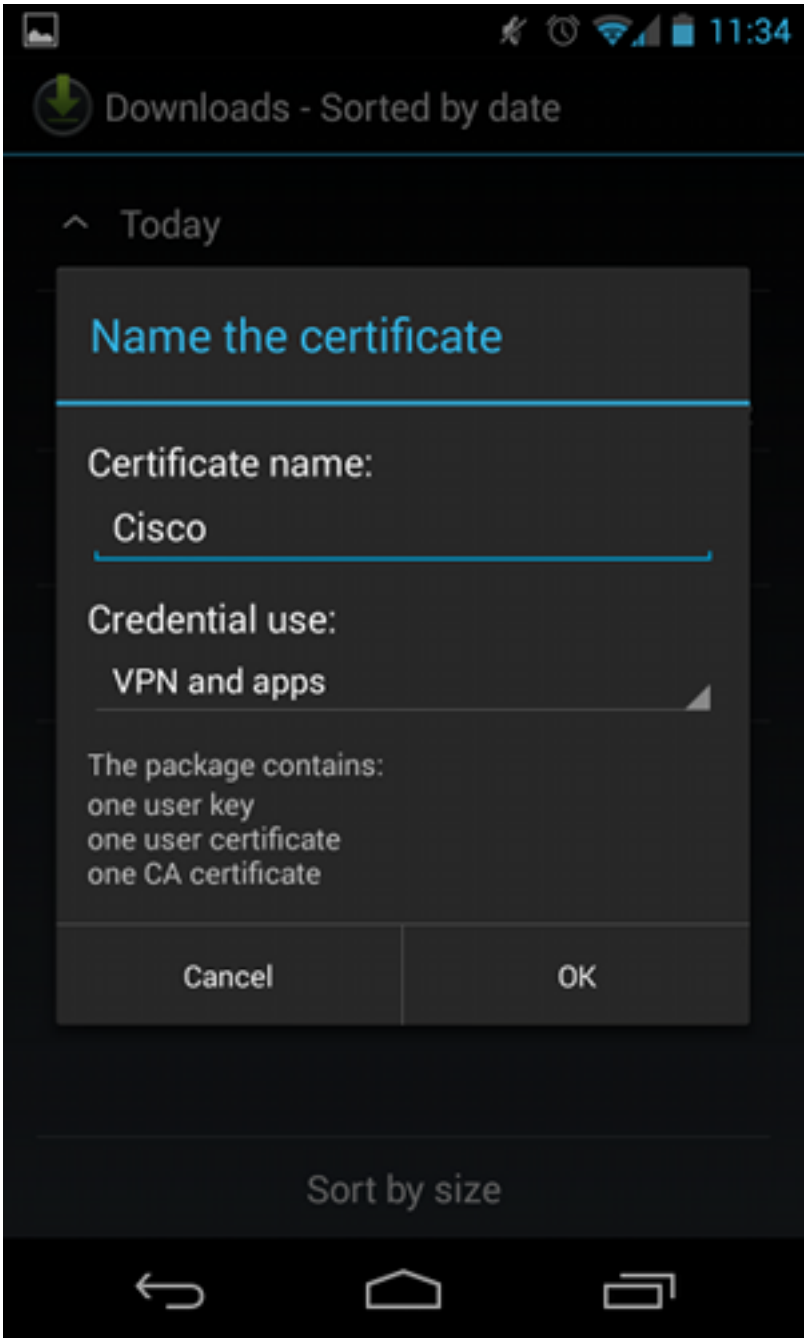
Cette procédure décrit comment installer les deux certificats :

1. Envoyez le fichier pfx par e-mail, puis ouvrez-le.

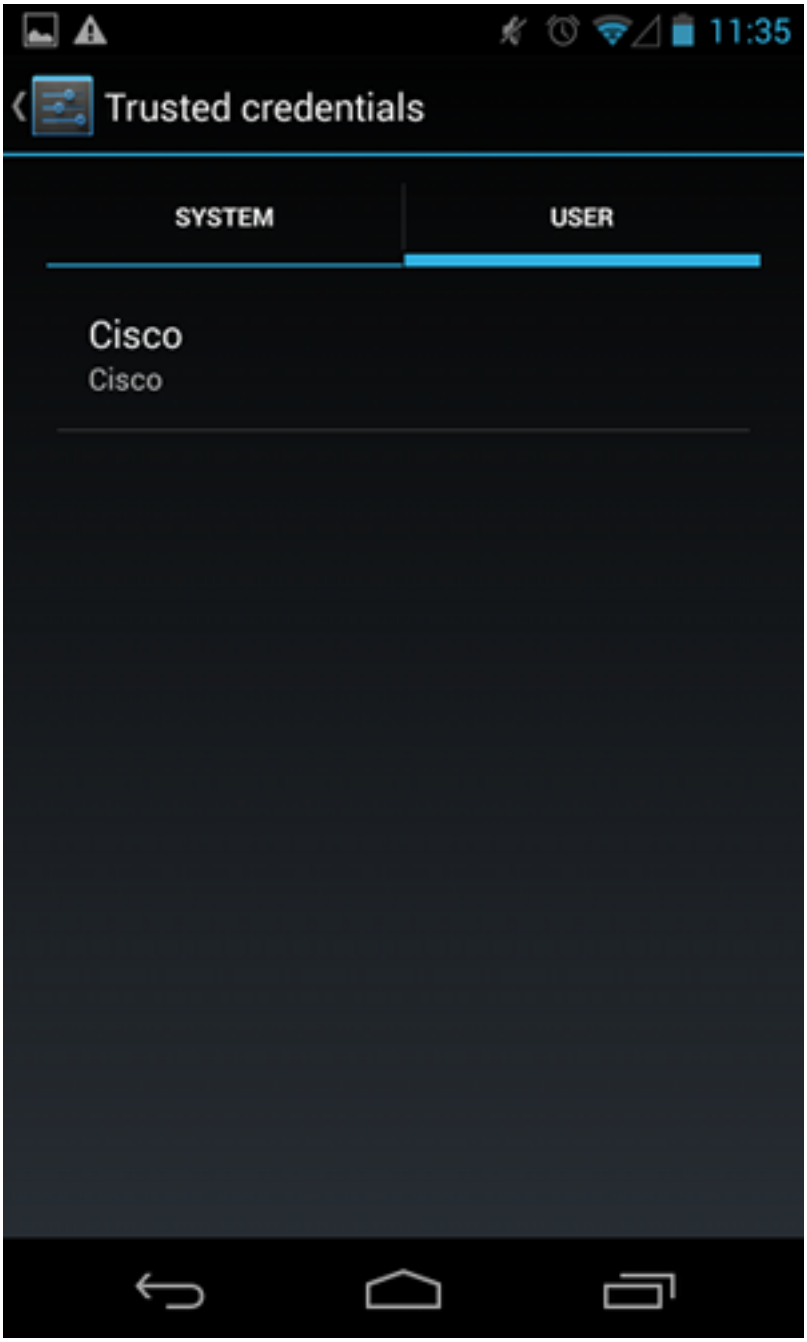
2. Indiquez le mot de passe utilisé lors de la génération du fichier pfx.



3. Indiquez le nom du certificat importé.



4. Accédez à **Paramètres > Sécurité > Informations d'identification de confiance** afin de vérifier l'installation du certificat. Le nouveau certificat doit apparaître dans le magasin d'utilisateurs :



Àce stade, un certificat utilisateur et un certificat CA sont installés. Le fichier pfx est un conteneur pkcs12 avec le certificat utilisateur et le certificat CA.

Android a des exigences précises lors de l'importation de certificats. Par exemple, pour qu'un certificat d'autorité de certification soit importé avec succès, Android nécessite que l'autorité de certification de contrainte de base de l'extension x509v3 soit définie sur TRUE. Par conséquent, lorsque vous générez une autorité de certification ou utilisez votre propre autorité de certification, il est important de vérifier qu'elle possède l'extension correcte :

```
pluton custom_ca # openssl x509 -in ca.crt -text
Certificate:
   Data&colon;
       Version: 3 (0x2)
       Serial Number:
           dc:8e:ad:98:72:3d:f5:6a
       Signature Algorithm: sha1WithRSAEncryption
       Issuer: C=PL, ST=Malopolskie, L=Krakow, O=Cisco, OU=Cisco TAC, CN=Cisco
<.....output omitted>
```

```
 X509v3 Basic Constraints:
                CA:TRUE

<.....output omitted>
```

Authentification EAP

Configuration du logiciel Cisco IOS pour l'authentification EAP

IKEv2 permet l'utilisation d'une pile de protocoles EAP afin d'exécuter l'authentification des utilisateurs. La passerelle VPN se présente avec le certificat. Une fois que le client fait confiance à ce certificat, il répond à l'identité de la demande EAP de la passerelle. Le logiciel Cisco IOS utilise cette identité et envoie un message Radius-Request au serveur AAA (Authentication, Authorization, and Accounting). Une session EAP-MD5 est établie entre le demandeur (Android) et le serveur d'authentification (Access Control Server [ACS] ou ISE).

Après une authentification EAP-MD5 réussie, comme indiqué par un message Radius-Accept, le logiciel Cisco IOS utilise le mode de configuration afin de transmettre l'adresse IP au client et de poursuivre la négociation du sélecteur de trafic.

Notez qu'Android a envoyé IKEID=cisco (tel que configuré). Cet IKEID reçu sur le logiciel Cisco IOS correspond à 'ikev2 profile PROF'.

```
aaa new-model
aaa authentication login eap-list-radius group radius
aaa authorization network IKE2_AUTHOR_LOCAL local

crypto pki trustpoint TP
 revocation-check none

crypto ikev2 authorization policy IKE2_AUTHOR_POLICY
 pool POOL
!
crypto ikev2 proposal ikev2-proposal
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy ikev2-policy
 proposal ikev2-proposal
!
!
crypto ikev2 profile PROF
 match identity remote key-id cisco
 authentication remote eap query-identity
 authentication local rsa-sig
 pki trustpoint TP
 aaa authentication eap eap-list-radius
 aaa authorization group eap list IKE2_AUTHOR_LOCAL IKE2_AUTHOR_POLICY
 aaa authorization user eap cached
 virtual-template 1

crypto ipsec transform-set 3DES-MD5 esp-aes esp-sha-hmac
 mode tunnel
```

```
!
crypto ipsec profile PROF
 set transform-set 3DES-MD5
 set ikev2-profile PROF

interface GigabitEthernet0/0
 ip address 10.48.64.15 255.255.255.128

interface Virtual-Template1 type tunnel
 ip unnumbered GigabitEthernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile PROF

ip local pool POOL 192.168.0.1 192.168.0.10

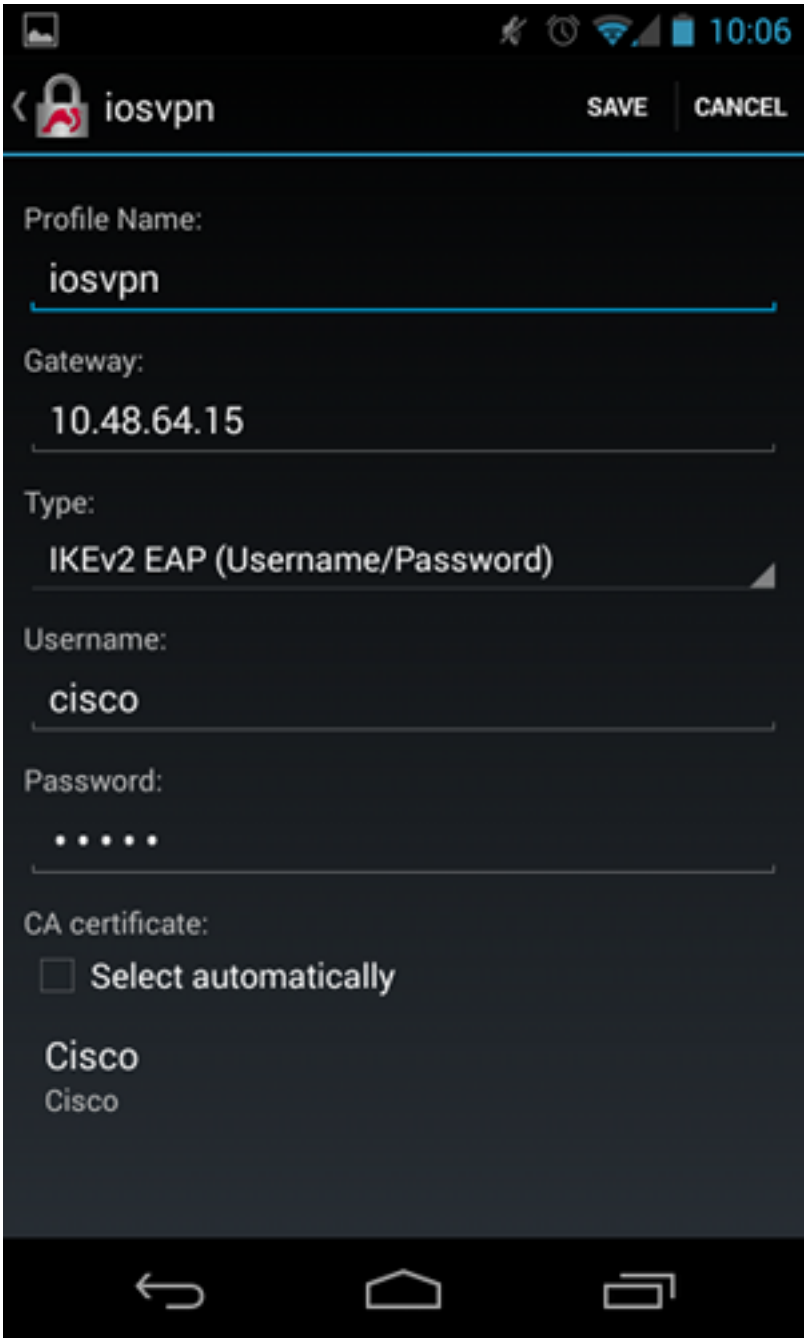
radius-server host 10.48.66.185 key cisco
```

Configuration Android pour l'authentification EAP

Android strongSwan doit avoir EAP configuré :

1. Désactiver la sélection automatique des certificats ; sinon, 100 CERT_REQ ou plus sont envoyés dans le troisième paquet.

2. Choisissez un certificat spécifique (CA) importé à l'étape précédente ; le nom d'utilisateur et le mot de passe doivent être identiques à ceux du serveur AAA.



Test d'authentification EAP

Dans le logiciel Cisco IOS, ce sont les débogages les plus importants pour l'authentification EAP. La plupart des résultats ont été omis par souci de clarté :

```
debug crypto ikev2 error
debug crypto ikev2 internal
debug radius authentication
debug radius verbose

IKEv2:New ikev2 sa request admitted
IKEv2:(SA ID = 1):Searching policy based on peer's identity 'cisco' of type
'FQDN'
IKEv2:(1): Choosing IKE profile PROF
IKEv2:Sending certificates as X509 certificates

RADIUS(00000025): Send Access-Request to 10.48.66.185:1645 id 1645/4,len 110
RADIUS: Received from id 1645/4 10.48.66.185:1645, Access-Challenge, len 79
```

```
RADIUS(00000025): Send Access-Request to 10.48.66.185:1645 id 1645/5,len 141
RADIUS: Received from id 1645/5 10.48.66.185:1645, Access-Challenge, len 100
RADIUS(00000025): Send Access-Request to 10.48.66.185:1645 id 1645/6,len 155
RADIUS: Received from id 1645/6 10.48.66.185:1645, Access-Accept, len 76

IKEv2:(SA ID = 1):SM Trace-> SA: I_SPI=AABAB198FACAAEDE R_SPI=D61F37C4DC875001
(R) MsgID = 00000004 CurState: R_PROC_EAP_RESP Event: EV_RECV_EAP_SUCCESS

IKEv2:IKEv2 local AAA author request for 'IKE2_AUTHOR_POLICY'
IKEv2:Received group author attributes: ipv4-pool: POOL, route-accept any tag:1
distance:1
IKEv2:Allocated addr 192.168.0.2 from local pool POOL
IKEv2:(SA ID = 1):SM Trace-> SA: I_SPI=AABAB198FACAAEDE R_SPI=D61F37C4DC875001
(R) MsgID = 00000005 CurState: R_VERIFY_AUTH Event:
   EV_OK_RECD_VERIFY_IPSEC_POLICY
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state
to up
```

Les journaux Android indiquent :

```
00[DMN] Starting IKE charon daemon (strongSwan 5.1.0dr2,
Linux 3.4.0-perf-gf43c3d9, armv7l)
00[KNL] kernel-netlink plugin might require CAP_NET_ADMIN capability
00[LIB] loaded plugins: androidbridge charon android-log openssl fips-prf
random nonce pubkey pkcs1 pkcs8 pem xcbc hmac socket-default kernel-netlink
00[LIB] unable to load 9 plugin features (9 due to unmet dependencies)
00[JOB] spawning 16 worker threads
13[IKE] initiating IKE_SA android[1] to 10.48.64.15
13[ENC] generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
13[NET] sending packet: from 10.147.24.153[45581] to 10.48.64.15[500]
(648 bytes)
11[NET] received packet: from 10.48.64.15[500] to 10.147.24.153[45581]
(497 bytes)
11[ENC] parsed IKE_SA_INIT response 0 [ SA KE No V V N(NATD_S_IP) N(NATD_D_IP)
CERTREQ N(HTTP_CERT_LOOK) ]
11[ENC] received unknown vendor ID:
43:49:53:43:4f:2d:44:45:4c:45:54:45:2d:52:45:41:53:4f:4e
11[ENC] received unknown vendor ID:
46:4c:45:58:56:50:4e:2d:53:55:50:50:4f:52:54:45:44
11[IKE] faking NAT situation to enforce UDP encapsulation
11[IKE] cert payload ANY not supported - ignored
11[IKE] sending cert request for "C=PL, ST=Malopolskie, L=Krakow, O=Cisco,
OU=Cisco TAC, CN=Cisco"
11[IKE] establishing CHILD_SA android
11[ENC] generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) CERTREQ
CP(ADDR ADDR6 DNS DNS6) N(ESP_TFC_PAD_N) SA TSi TSr N(MOBIKE_SUP)
11[NET] sending packet: from 10.147.24.153[35564] to 10.48.64.15[4500]
(508 bytes)
10[NET] received packet: from 10.48.64.15[4500] to 10.147.24.153[35564]
(1292 bytes)
10[ENC] parsed IKE_AUTH response 1 [ V IDr CERT AUTH EAP/REQ/ID ]
10[IKE] received end entity cert "C=PL, ST=Malopolska, L=Krakow, O=Cisco,
OU=TAC, CN=IOS"
10[CFG] using certificate "C=PL, ST=Malopolska, L=Krakow, O=Cisco, OU=TAC,
CN=IOS"
10[CFG] using trusted ca certificate "C=PL, ST=Malopolskie, L=Krakow, O=Cisco,
OU=Cisco TAC, CN=Cisco"
10[CFG] reached self-signed root ca with a path length of 0
10[IKE] authentication of '10.48.64.15' with RSA signature successful
10[IKE] server requested EAP_IDENTITY (id 0x3B), sending 'cisco'
10[ENC] generating IKE_AUTH request 2 [ EAP/RES/ID ]
10[NET] sending packet: from 10.147.24.153[35564] to 10.48.64.15[4500]
(76 bytes)
```

```
09[NET] received packet: from 10.48.64.15[4500] to 10.147.24.153[35564]
(76 bytes)
09[ENC] parsed IKE_AUTH response 2 [ EAP/REQ/TLS ]
09[IKE] server requested EAP_TLS authentication (id 0x59)
09[IKE] EAP method not supported, sending EAP_NAK
09[ENC] generating IKE_AUTH request 3 [ EAP/RES/NAK ]
09[NET] sending packet: from 10.147.24.153[35564] to 10.48.64.15[4500]
(76 bytes)
08[NET] received packet: from 10.48.64.15[4500] to 10.147.24.153[35564]
(92 bytes)
08[ENC] parsed IKE_AUTH response 3 [ EAP/REQ/MD5 ]
08[IKE] server requested EAP_MD5 authentication (id 0x5A)
08[ENC] generating IKE_AUTH request 4 [ EAP/RES/MD5 ]
08[NET] sending packet: from 10.147.24.153[35564] to 10.48.64.15[4500]
(92 bytes)
07[NET] received packet: from 10.48.64.15[4500] to 10.147.24.153[35564]
(76 bytes)
07[ENC] parsed IKE_AUTH response 4 [ EAP/SUCC ]
07[IKE] EAP method EAP_MD5 succeeded, no MSK established
07[IKE] authentication of 'cisco' (myself) with EAP
07[ENC] generating IKE_AUTH request 5 [ AUTH ]
07[NET] sending packet: from 10.147.24.153[35564] to 10.48.64.15[4500]
(92 bytes)
06[NET] received packet: from 10.48.64.15[4500] to 10.147.24.153[35564]
(236 bytes)
06[ENC] parsed IKE_AUTH response 5 [ AUTH CP(ADDR) SA TSi TSr N(SET_WINSIZE)
N(ESP_TFC_PAD_N) N(NON_FIRST_FRAG) ]
06[IKE] authentication of '10.48.64.15' with EAP successful
06[IKE] IKE_SA android[1] established between
10.147.24.153[cisco]...10.48.64.15[10.48.64.15]
06[IKE] scheduling rekeying in 35421s
06[IKE] maximum IKE_SA lifetime 36021s
06[IKE] installing new virtual IP 192.168.0.1
06[IKE] received ESP_TFC_PADDING_NOT_SUPPORTED, not using ESPv3 TFC padding
06[IKE] CHILD_SA android{1} established with SPIs c776cb4f_i ea27f072_o and
TS 192.168.0.1/32 === 0.0.0.0/0
06[DMN] setting up TUN device for CHILD_SA android{1}
06[DMN] successfully created TUN device
```

Cet exemple montre comment vérifier l'état du logiciel Cisco IOS :

```
BSAN-2900-1#show crypto session detail
Crypto session current status

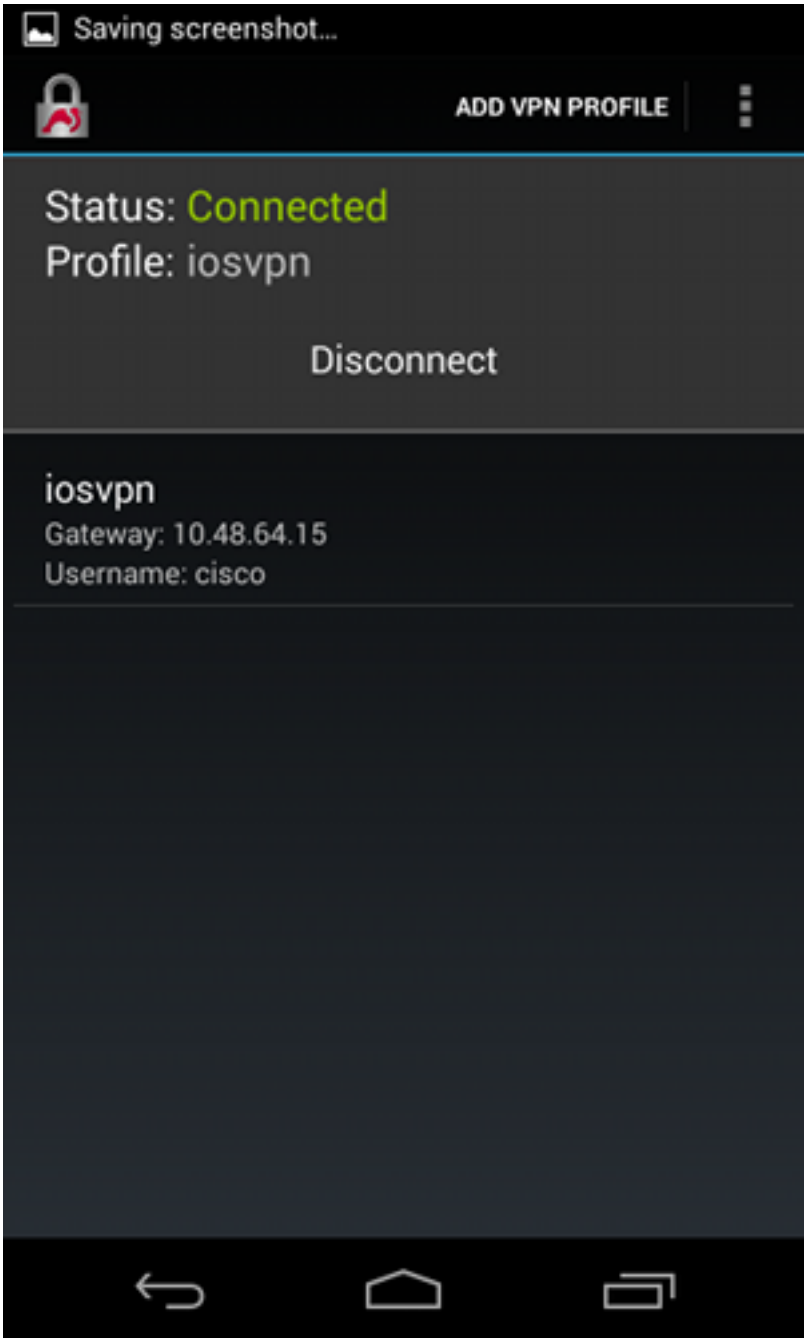
Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Virtual-Access1
Uptime: 00:02:12
Session status: UP-ACTIVE
Peer: 10.147.24.153 port 60511 fvrf: (none) ivrf: (none)
     Phase1_id: cisco
     Desc: (none)
 IKEv2 SA: local 10.48.64.15/4500 remote 10.147.24.153/60511 Active
         Capabilities:NX connid:1 lifetime:23:57:48
 IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 host 192.168.0.2
       Active SAs: 2, origin: crypto map
       Inbound:  #pkts dec'ed 40 drop 0 life (KB/Sec) 4351537/3468
       Outbound: #pkts enc'ed 5 drop 0 life (KB/Sec) 4351542/3468

BSAN-2900-1#show crypto ikev2 sa detailed
 IPv4 Crypto IKEv2  SA
```

```
Tunnel-id Local                 Remote                fvrf/ivrf         Status
1         10.48.64.15/4500      10.147.24.153/60511   none/none         READY
     Encr: AES-CBC, keysize: 128, Hash: SHA96, DH Grp:14, Auth sign: RSA,
Auth verify: EAP
     Life/Active Time: 86400/137 sec
     CE id: 1002, Session-id: 2
     Status Description: Negotiation done
     Local spi: D61F37C4DC875001       Remote spi: AABAB198FACAAEDE
     Local id: 10.48.64.15
     Remote id: cisco
     Remote EAP id: cisco
     Local req msg id:  0              Remote req msg id:  6
     Local next msg id: 0              Remote next msg id: 6
     Local req queued:  0              Remote req queued:  6
     Local window:      5              Remote window:      1
     DPD configured for 0 seconds, retry 0
     Fragmentation not configured.
     Extended Authentication configured.
     NAT-T is detected  outside
     Cisco Trust Security SGT is disabled
     Assigned host addr: 192.168.0.2
     Initiator of SA : No
```

Ces figures montrent comment vérifier l'état d'Android :


Saving screenshot...
ADD VPN PROFILE
Status: Connected
Profile: iosvpn
Disconnect
iosvpn
Gateway: 10.48.64.15
Username: cisco



Authentification RSA

Configuration du logiciel Cisco IOS pour l'authentification RSA

Dans l'authentification Rivest-Shamir-Adleman (RSA), Android envoie le certificat afin de s'authentifier auprès du logiciel Cisco IOS. C'est pourquoi la carte de certificat qui lie ce trafic à un profil IKEv2 spécifique est nécessaire. L'authentification EAP utilisateur n'est pas requise.

Voici un exemple de définition de l'authentification RSA pour un homologue distant :

```
crypto pki certificate map CERT_MAP 10
 subject-name co android

crypto ikev2 profile PROF
 match certificate CERT_MAP
```

```
authentication remote rsa-sig
authentication local rsa-sig
pki trustpoint TP
aaa authorization group cert list IKE2_AUTHOR_LOCAL IKE2_AUTHOR_POLICY
virtual-template 1
```

Configuration Android pour authentification RSA

Les informations d'identification de l'utilisateur ont été remplacées par le certificat utilisateur :



Test d'authentification RSA

Dans le logiciel Cisco IOS, ce sont les débogages les plus importants pour l'authentification RSA. La plupart des résultats ont été omis par souci de clarté :

```
debug crypto ikev2 error
```

```
debug crypto ikev2 internal
debug crypto pki transactions
debug crypto pki validation
debug crypto pki messages

IKEv2:New ikev2 sa request admitted
IKEv2:(SA ID = 1):Searching policy based on peer's identity 'cn=android,ou=TAC,
o=Cisco,l=Krakow,st=Malopolska,c=PL' of type 'DER ASN1 DN'
IKEv2:(1): Choosing IKE profile PROF
IKEv2:Sending certificates as X509 certificates
IKEv2:(SA ID = 1):Peer's authentication method is 'RSA'
IKEv2:Peer has sent X509 certificates
CRYPTO_PKI: Found a issuer match
CRYPTO_PKI: (9000B) Certificate is verified
CRYPTO_PKI: (9000B) Certificate validation succeeded
IKEv2:(SA ID = 1):[Crypto Engine -> IKEv2] Verification of signed
authentication data PASSED

IKEv2:IKEv2 local AAA author request for 'IKE2_AUTHOR_POLICY'
IKEv2:Received group author attributes: ipv4-pool: POOL, route-accept any tag:1
distance:1
IKEv2:Allocated addr 192.168.0.3 from local pool POOL
IKEv2:(SA ID = 1):SM Trace-> SA: I_SPI=E53A57E359A8437C R_SPI=A03D273FC75EEBD9
(R) MsgID = 00000001 CurState: R_VERIFY_AUTH Event:
EV_OK_RECD_VERIFY_IPSEC_POLICY
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state
to up
```

Les journaux Android indiquent :

```
00[DMN] Starting IKE charon daemon (strongSwan 5.1.0dr2,
Linux 3.4.0-perf-gf43c3d9, armv7l)
00[KNL] kernel-netlink plugin might require CAP_NET_ADMIN capability
00[LIB] loaded plugins: androidbridge charon android-log openssl fips-prf
random nonce pubkey pkcs1 pkcs8 pem xcbc hmac socket-default
00[LIB] unable to load 9 plugin features (9 due to unmet dependencies)
00[JOB] spawning 16 worker threads
05[CFG] loaded user certificate 'C=PL, ST=Malopolska, L=Krakow, O=Cisco,
OU=TAC, CN=android' and private key
05[CFG] loaded CA certificate 'C=PL, ST=Malopolskie, L=Krakow, O=Cisco,
OU=Cisco TAC, CN=Cisco'

05[IKE] initiating IKE_SA android[4] to 10.48.64.15
05[ENC] generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
05[NET] sending packet: from 10.147.24.153[34697] to 10.48.64.15[500]
(648 bytes)
10[NET] received packet: from 10.48.64.15[500] to 10.147.24.153[34697]
(497 bytes)
10[ENC] parsed IKE_SA_INIT response 0 [ SA KE No V V N(NATD_S_IP) N(NATD_D_IP)
CERTREQ N(HTTP_CERT_LOOK) ]
10[ENC] received unknown vendor ID:
43:49:53:43:4f:2d:44:45:4c:45:54:45:2d:52:45:41:53:4f:4e
10[ENC] received unknown vendor ID:
46:4c:45:58:56:50:4e:2d:53:55:50:50:4f:52:54:45:44
10[IKE] faking NAT situation to enforce UDP encapsulation
10[IKE] cert payload ANY not supported - ignored
10[IKE] sending cert request for "C=PL, ST=Malopolskie, L=Krakow, O=Cisco,
OU=Cisco TAC, CN=Cisco"
10[IKE] authentication of 'C=PL, ST=Malopolska, L=Krakow, O=Cisco, OU=TAC,
CN=android' (myself) with RSA signature successful
10[IKE] sending end entity cert "C=PL, ST=Malopolska, L=Krakow, O=Cisco,
OU=TAC, CN=android"
10[IKE] establishing CHILD_SA android
```

```
10[ENC] generating IKE_AUTH request 1 [ IDi CERT N(INIT_CONTACT) CERTREQ
AUTH CP(ADDR ADDR6 DNS DNS6) N(ESP_TFC_PAD_N) SA
10[NET] sending packet: from 10.147.24.153[44527] to 10.48.64.15[4500]
(1788 bytes)
12[NET] received packet: from 10.48.64.15[4500] to 10.147.24.153[44527]
(1420 bytes)
12[ENC] parsed IKE_AUTH response 1 [ V IDr CERT AUTH CP(ADDR) SA TSi TSr
N(SET_WINSIZE) N(ESP_TFC_PAD_N) N(NON_FIRST_FRAG)
12[IKE] received end entity cert "C=PL, ST=Malopolska, L=Krakow, O=Cisco,
OU=TAC, CN=IOS"
12[CFG] using certificate "C=PL, ST=Malopolska, L=Krakow, O=Cisco, OU=TAC,
CN=IOS"
12[CFG] using trusted ca certificate "C=PL, ST=Malopolskie, L=Krakow, O=Cisco,
OU=Cisco TAC, CN=Cisco"
12[CFG] reached self-signed root ca with a path length of 0
12[IKE] authentication of '10.48.64.15' with RSA signature successful
12[IKE] IKE_SA android[4] established between 10.147.24.153[C=PL,
ST=Malopolska, L=Krakow, O=Cisco, OU=TAC,
CN=android]...10.48.64.15[10.48.64.15]
12[IKE] scheduling rekeying in 35413s
12[IKE] maximum IKE_SA lifetime 36013s
12[IKE] installing new virtual IP 192.168.0.3
12[IKE] received ESP_TFC_PADDING_NOT_SUPPORTED, not using ESPv3 TFC padding
12[IKE] CHILD_SA android{4} established with SPIs ecb3af87_i b2279175_o and
TS 192.168.0.3/32 === 0.0.0.0/0
12[DMN] setting up TUN device for CHILD_SA android{4}
12[DMN] successfully created TUN device
```

Dans le logiciel Cisco IOS, RSA est utilisé pour la signature et la vérification ; dans le scénario précédent, le PAE a été utilisé pour la vérification :

```
BSAN-2900-1#show crypto ikev2 sa detailed
 IPv4 Crypto IKEv2  SA

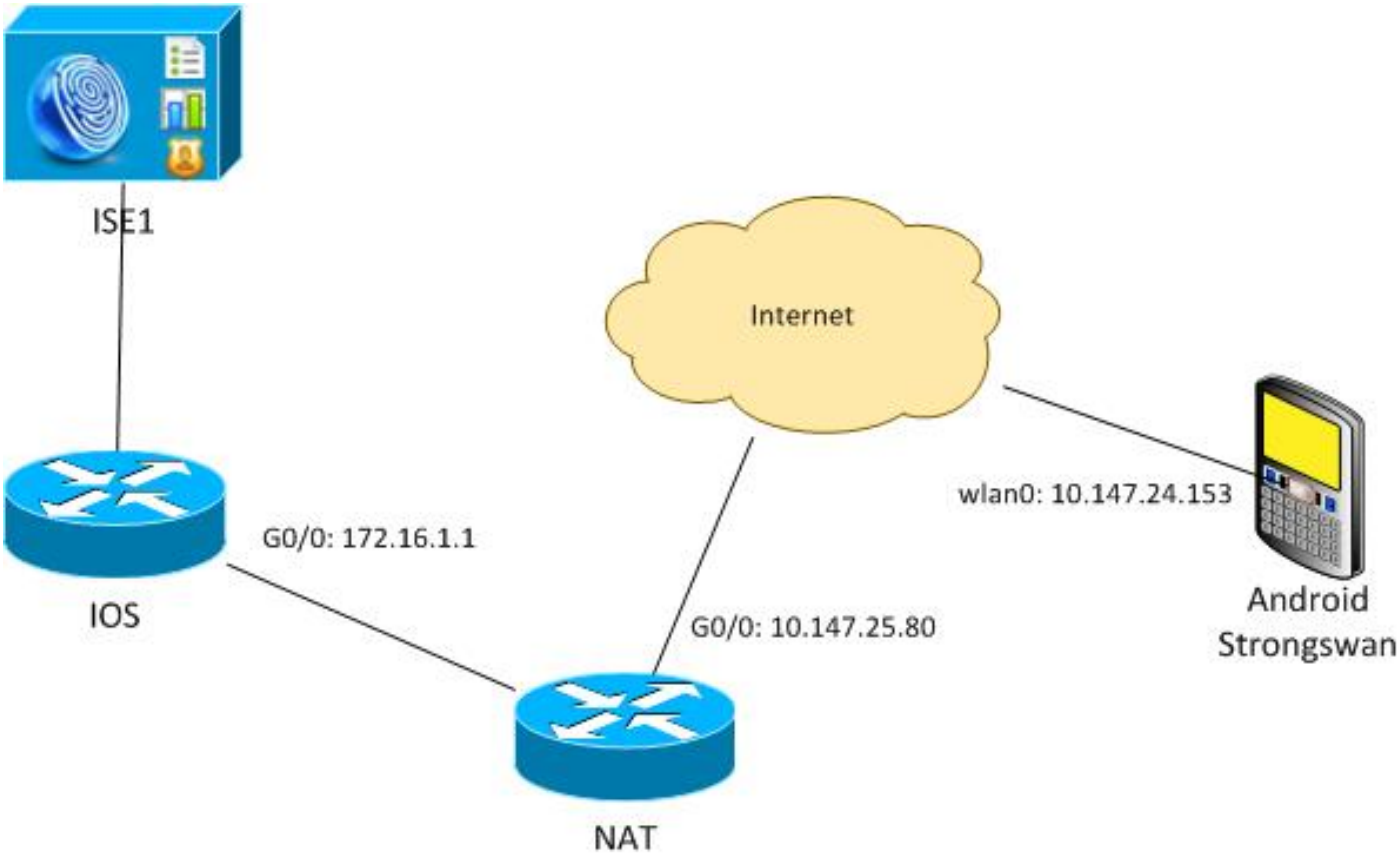
Tunnel-id Local                 Remote                fvrf/ivrf          Status
1         10.48.64.15/4500      10.147.24.153/44527   none/none          READY
     Encr: AES-CBC, keysize: 128, Hash: SHA96, DH Grp:14, Auth sign: RSA,
Auth verify: RSA
     Life/Active Time: 86400/16 sec
     CE id: 1010, Session-id: 3
     Status Description: Negotiation done
     Local spi: A03D273FC75EEBD9       Remote spi: E53A57E359A8437C
     Local id: 10.48.64.15
     Remote id: cn=android,ou=TAC,o=Cisco,l=Krakow,st=Malopolska,c=PL
     Local req msg id:  0              Remote req msg id:  2
     Local next msg id: 0              Remote next msg id: 2
     Local req queued:  0              Remote req queued:  2
     Local window:      5              Remote window:      1
     DPD configured for 0 seconds, retry 0
     Fragmentation not configured.
     Extended Authentication not configured.
     NAT-T is detected  outside
     Cisco Trust Security SGT is disabled
     Assigned host addr: 192.168.0.3
     Initiator of SA : No
```

La vérification de l'état sur Android est similaire à celle du scénario précédent.

Passerelle VPN derrière la NAT - fortes limitations du logiciel Cisco IOS et de strongSwan

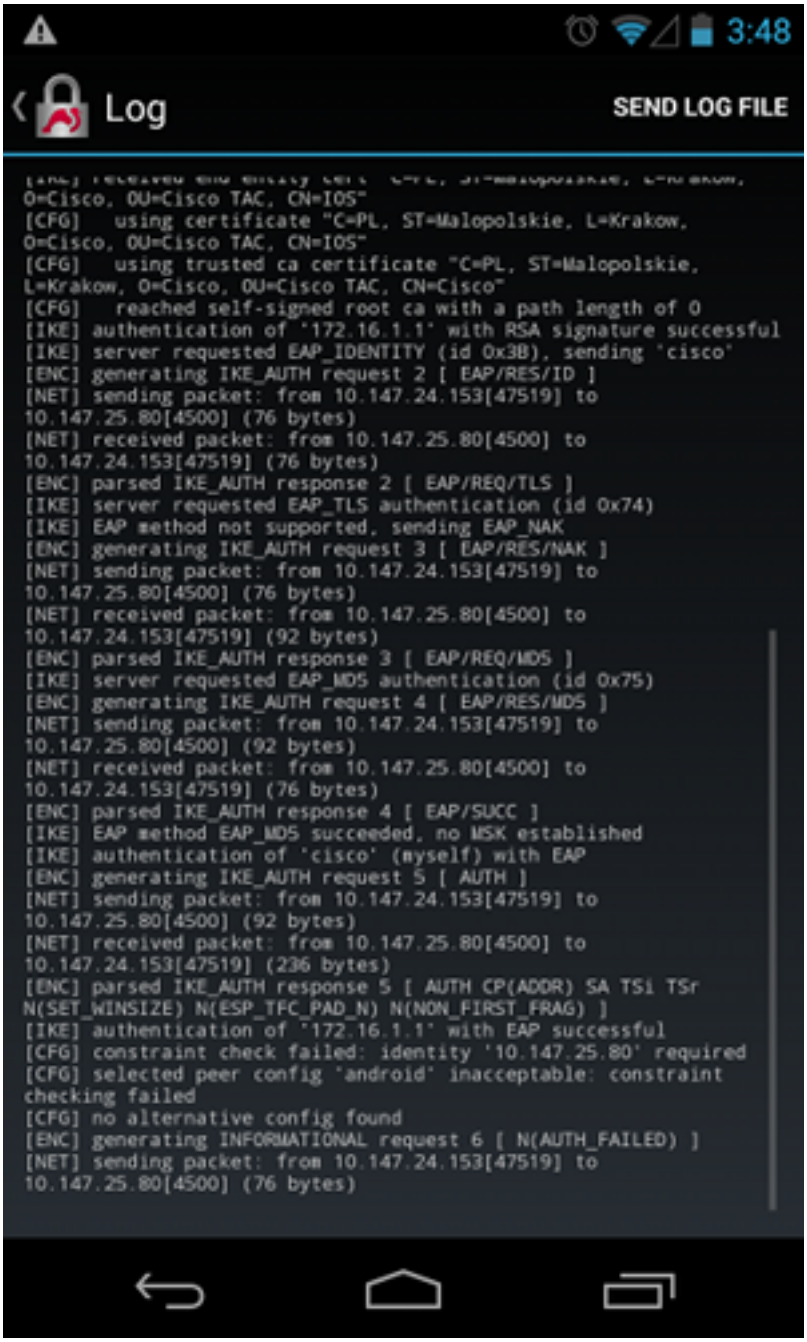
Cet exemple explique une limitation des vérifications de certificat strongSwan.

Supposons que l'adresse IP de la passerelle VPN du logiciel Cisco IOS est traduite de manière statique de 172.16.1.1 à 10.147.25.80. L'authentification EAP est utilisée.



Supposez également que le certificat du logiciel Cisco IOS a un autre nom de sujet pour 172.16.1.1 et 10.147.25.80.

Après une authentification EAP réussie, Android effectue la vérification et tente de trouver l'adresse IP de l'homologue qui a été utilisé dans la configuration Android (10.147.25.80) dans l'extension Subject Alternative Name. La vérification échoue :

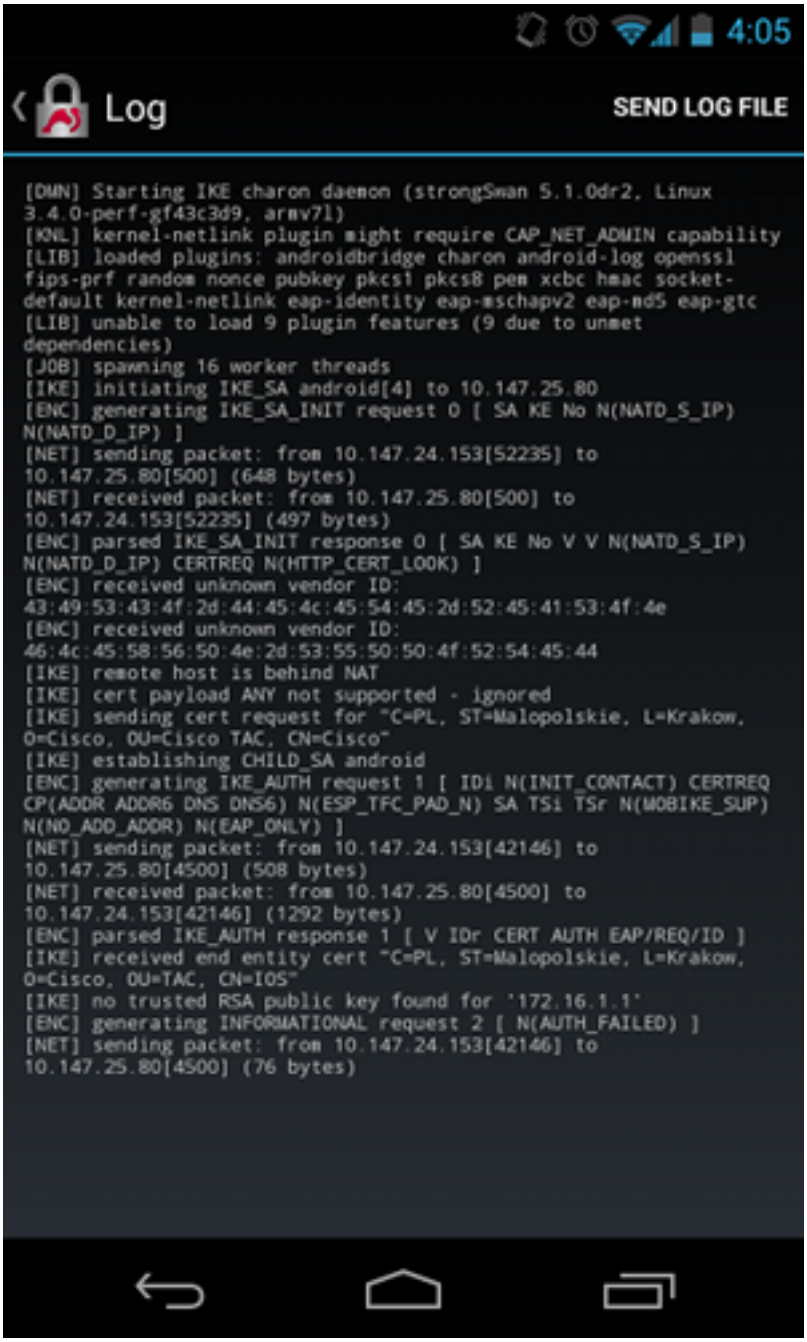


Les journaux indiquent :

```
constraint check failed: identity '10.147.25.80' required
```

L'échec s'est produit car Android ne peut lire que la première extension Subject Alternative Name (172.16.1.1).

Maintenant, supposez que le certificat du logiciel Cisco IOS a les deux adresses dans Nom alternatif de l'objet mais dans l'ordre inverse : 10.147.25.80 et 172.16.1.1. Android effectue la validation lorsqu'il reçoit l'IKEID, qui est l'adresse IP de la passerelle VPN (172.16.1.1), dans le troisième paquet :



Le journal indique maintenant :

```
no trusted RSA public key found for '172.16.1.1'
```

Par conséquent, lorsqu'Android reçoit l'IKEID, il doit trouver l'IKEID dans le champ Subject Alternative Name et ne peut utiliser que la première adresse IP.

> **Note**: Dans l'authentification EAP, l'IKEID envoyé par le logiciel Cisco IOS est l'adresse IP par défaut. Dans l'authentification RSA, l'IKEID est le DN du certificat par défaut. Utilisez la commande **identity** sous le profil ikev2 afin de modifier ces valeurs manuellement.

Vérification

Les procédures de vérification et de test sont disponibles dans les exemples de configuration.

Dépannage

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

strongSwan CA Multiple CERT_REQ

Lorsque le paramètre de certificat sur strongSwan est Automatic Selection (Sélection automatique) (valeur par défaut), Android envoie CERT_REQ pour tous les certificats de confiance dans le magasin local du troisième paquet . Le logiciel Cisco IOS peut abandonner la demande car il reconnaît un grand nombre de demandes de certificat comme une attaque de déni de service :

```
*Jul 15 07:54:13: IKEv2:number of cert req exceeds the reasonable limit (100)
```

Source du tunnel sur DVTI

Bien qu'il soit assez courant de définir la source du tunnel sur une interface de tunnel virtuel (VTI), il n'est pas nécessaire ici. Supposons que la commande **tunnel source** est sous un VTI dynamique (DVTI) :

```
interface Virtual-Template1 type tunnel
 ip unnumbered GigabitEthernet0/0
 tunnel source GigabitEthernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile PROF
```

Après authentification, si le logiciel Cisco IOS tente de créer une interface d'accès virtuel clonée à partir d'un modèle virtuel, il renvoie une erreur :

```
*Aug  1 13:34:22 IKEv2:Allocated addr 192.168.0.9 from local pool POOL
*Aug  1 13:34:22 IKEv2:(SA ID = 1):Set received config mode data
*Aug  1 13:34:22 IKEv2:% DVTI create request sent for profile PROF with PSH
index 1
*Aug  1 13:34:22 IKEv2:Failed to process KMI delete SA message with error 4
*Aug  1 13:34:24 IKEv2:Got a packet from dispatcher
*Aug  1 13:34:24 IKEv2:Processing an item off the pak queue
*Aug  1 13:34:24 IKEv2:Negotiation context locked currently in use
```

Deux secondes après la panne, le logiciel Cisco IOS reçoit une nouvelle IKE_AUTH d'Android. Ce paquet est abandonné.

Demandes d'amélioration et de bogues du logiciel Cisco IOS

- ID de bogue Cisco CSCui46418, « Adresse IP IOS Ikev2 envoyée en tant qu'identité pour l'authentification RSA. »
  Ce bogue n'est pas un problème, tant que strongSwan peut voir un bon Subject Alternative Name (l'adresse IP) lorsqu'il recherche l'IKEID dans le certificat afin d'effectuer la vérification.
- Identifiant de bogue Cisco CSCui44976, « L'ICP IOS n'affichait pas correctement le nom alternatif de l'extension X509v3. »

Ce bogue se produit uniquement lorsqu'il y a plusieurs adresses IP dans le champ Subject Alternative Name. Seule la dernière adresse IP est affichée, mais cela n'a pas d'incidence sur l'utilisation des certificats. L'ensemble du certificat est envoyé et traité correctement.

- ID de bogue Cisco CSCui44783, « IOS ENH PKI capacité à générer CSR avec l'extension subject-alt-name. »
- ID de bogue Cisco CSCui44335, « Extensions x509 du certificat ENH ASA ».

Informations connexes

- Guide de configuration du VPN Cisco IOS 15.3
- Référence des commandes de Cisco IOS 15.3
- Guide de configuration de Cisco IOS Flex VPN
- Support et documentation techniques - Cisco Systems

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.