IKEv2 da Android strongSwan a Cisco IOS con autenticazione EAP e RSA

Sommario

Introduzione

In questo documento viene descritto come configurare la versione per dispositivi mobili di strongSwan per accedere a un gateway VPN con software Cisco IOS® tramite il protocollo Internet Key Exchange versione 2 (IKEv2).

Vengono presentati tre esempi:

- Telefono Android con strongSwan che si connette al gateway VPN del software Cisco IOS con autenticazione Extensible Authentication Protocol - Message Digest 5 (EAP-MD5).
- Telefono Android con strongSwan che si connette al gateway VPN del software Cisco IOS

con autenticazione certificato (RSA).
- Telefono Android con strongSwan che si connette al gateway VPN del software Cisco IOS dietro a Network Address Translation (NAT). È necessario che nel certificato gateway VPN siano presenti due estensioni x509 con nome alternativo soggetto.

Sono inclusi anche il software Cisco IOS e le limitazioni strongSwan.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Conoscenze base della configurazione di OpenSSL
- Conoscenze base della configurazione dell'interfaccia della riga di comando (CLI) del software Cisco IOS
- Conoscenze base di IKEv2

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

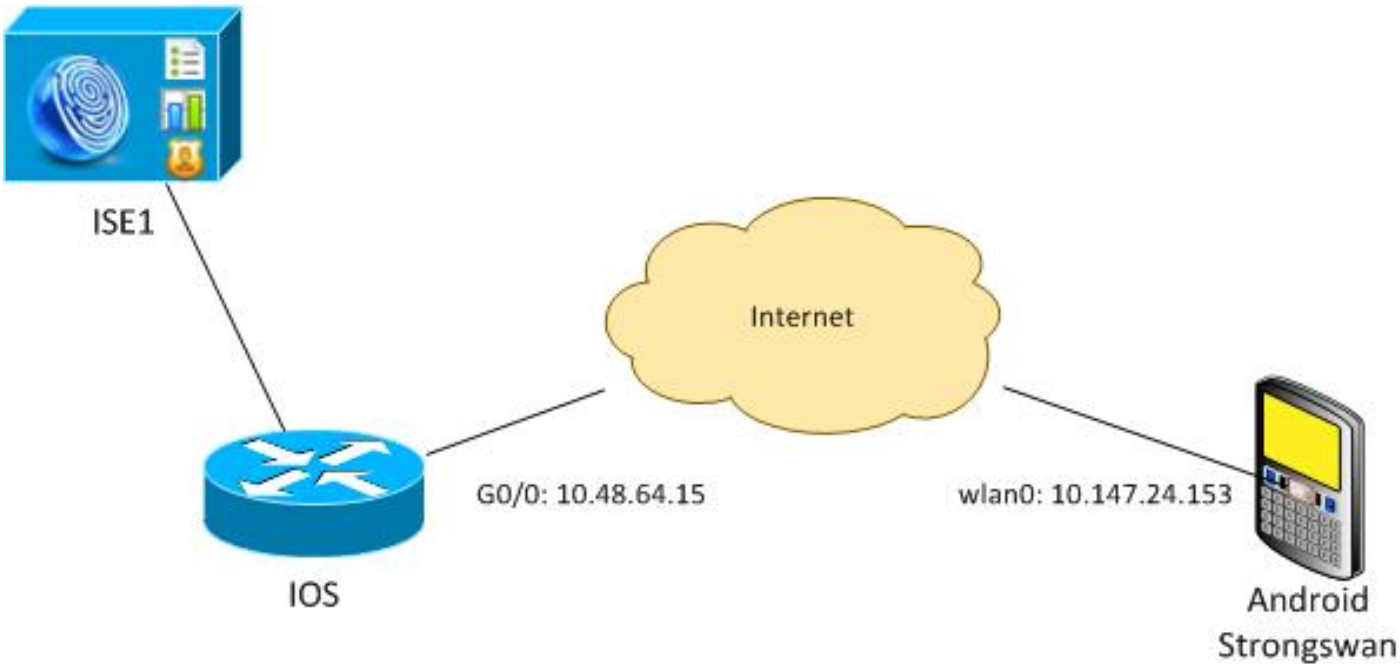
- Android 4.0 o versioni successive con strongSwan
- Software Cisco IOS release 15.3T o successive
- Software Cisco Identity Services Engine (ISE), versione 1.1.4 e successive

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Configurazione

> **Note:**
>
> Lo strumento Output Interpreter (solo utenti registrati) supporta alcuni comandi show. Usare lo strumento Output Interpreter per visualizzare un'analisi dell'output del comando **show.**
>
> consultare le informazioni importanti sui comandi di debug prima di usare i comandi di **debug.**

Esempio di rete



Android strongSwan stabilisce un tunnel IKEv2 con un gateway software Cisco IOS per accedere in modo sicuro alle reti interne.

Registrazione certificato

I certificati sono un prerequisito per l'autenticazione basata su EAP e su RSA.

Nello scenario di autenticazione EAP, un certificato è necessario solo sul gateway VPN. Il client si connette al software Cisco IOS solo quando il software presenta un certificato firmato da un'Autorità di certificazione (CA) considerata attendibile su Android. Viene quindi avviata una sessione EAP per consentire al client di autenticarsi al software Cisco IOS.

Per l'autenticazione basata su RSA, entrambi gli endpoint devono avere un certificato corretto.

Quando si utilizza un indirizzo IP come ID peer, sono previsti requisiti aggiuntivi per il certificato. Android strongSwan verifica se l'indirizzo IP del gateway VPN è incluso nell'estensione x509 Subject Alternative Name. In caso contrario, Android interrompe la connessione; si tratta di una buona pratica e di una raccomandazione della RFC 6125.

OpenSSL viene utilizzato come CA perché il software Cisco IOS ha una limitazione: impossibile generare certificati con un'estensione che include un indirizzo IP. Tutti i certificati vengono generati da OpenSSL e importati in Android e nel software Cisco IOS.

Nel software Cisco IOS, il comando **subject-alt-name** può essere usato per creare un'estensione che include un indirizzo IP, ma il comando funziona solo con certificati autofirmati. Cisco Bug ID CSCui44783, "IOS ENH PKI ability to generate CSR with subject-alt-name extension", è una richiesta di miglioramento che consente al software Cisco IOS di generare l'estensione per tutti i tipi di registrazione.

Questo è un esempio dei comandi che generano una CA:

```
#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 è un file di configurazione. L'estensione CA deve essere impostata su 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
```

I comandi che generano un certificato sono molto simili per il software Cisco IOS e Android. L'esempio presuppone che esista già una CA utilizzata per firmare il certificato:

```
#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 è un file di configurazione. L'estensione Nome soggetto alternativo è un'impostazione chiave. In questo esempio, l'estensione CA è impostata su 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
```

Ènecessario generare un certificato sia per il software Cisco IOS che per Android.

L'indirizzo IP 10.48.64.15 appartiene al gateway software Cisco IOS. Quando si genera un certificato per il software Cisco IOS, accertarsi che subjectAltName sia impostato su 10.48.64.15. Android convalida il certificato ricevuto dal software Cisco IOS e cerca di trovare il proprio indirizzo IP in subjectAltName.

Software Cisco IOS

Ènecessario che sul software Cisco IOS sia installato un certificato corretto per l'autenticazione basata su RSA e su EAP.

Il file pfx (che è un contenitore pkcs12) per il software Cisco IOS può essere importato:

```
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.
```

Per verificare che l'importazione sia stata completata, usare il comando **show crypto pki certificates verbose**:

```
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

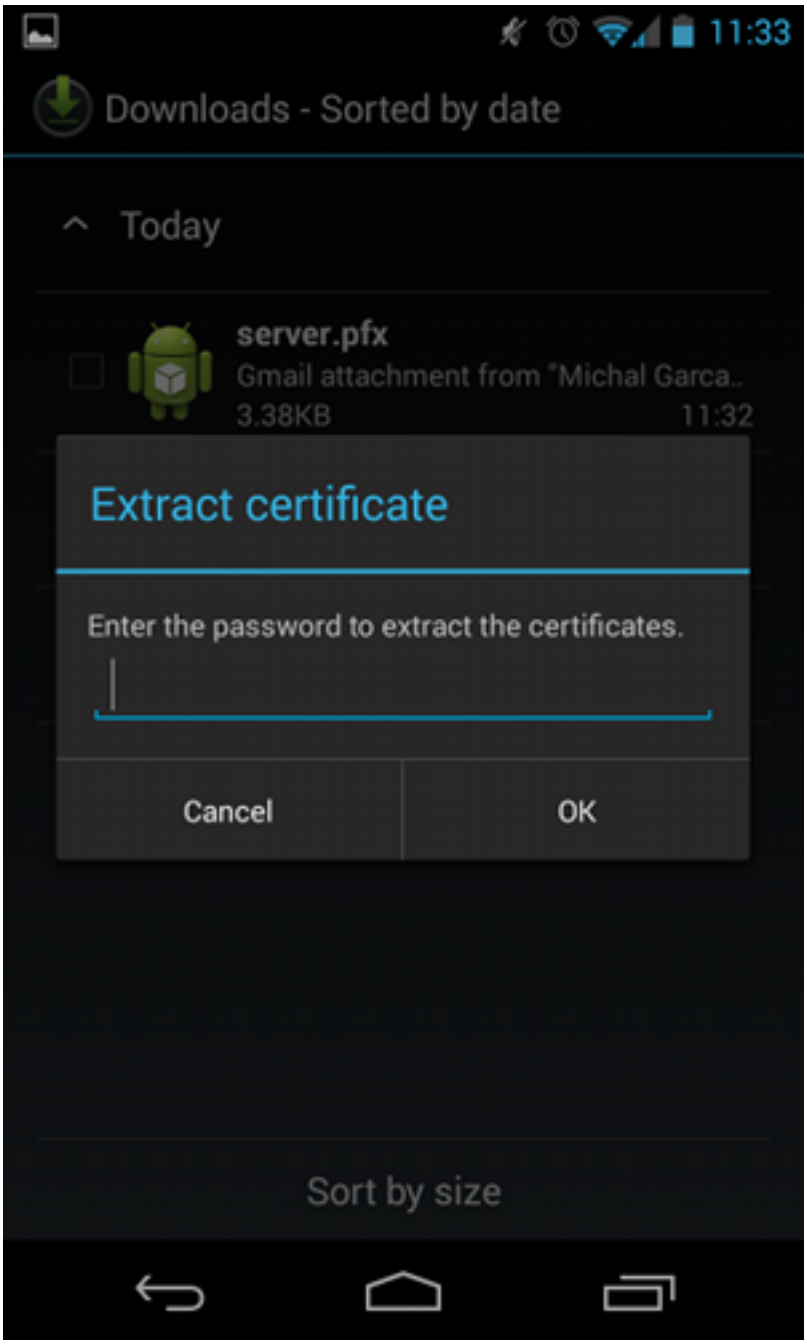
Per l'autenticazione basata su EAP, in Andorid deve essere installato solo il certificato CA corretto.

Per l'autenticazione basata su RSA, Andorid deve disporre sia del certificato CA che del proprio certificato.

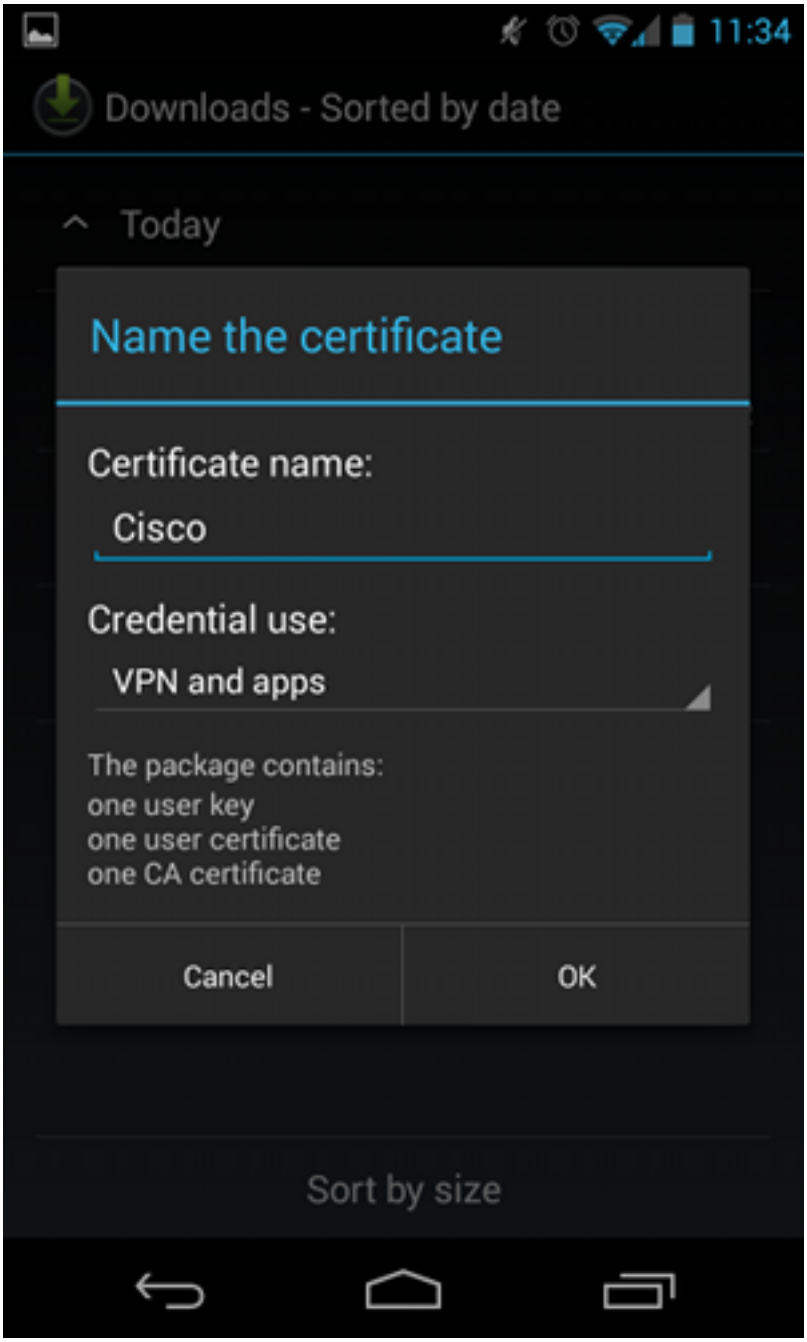
In questa procedura viene descritto come installare entrambi i certificati:

1. Inviare il file pfx tramite posta elettronica e aprirlo.

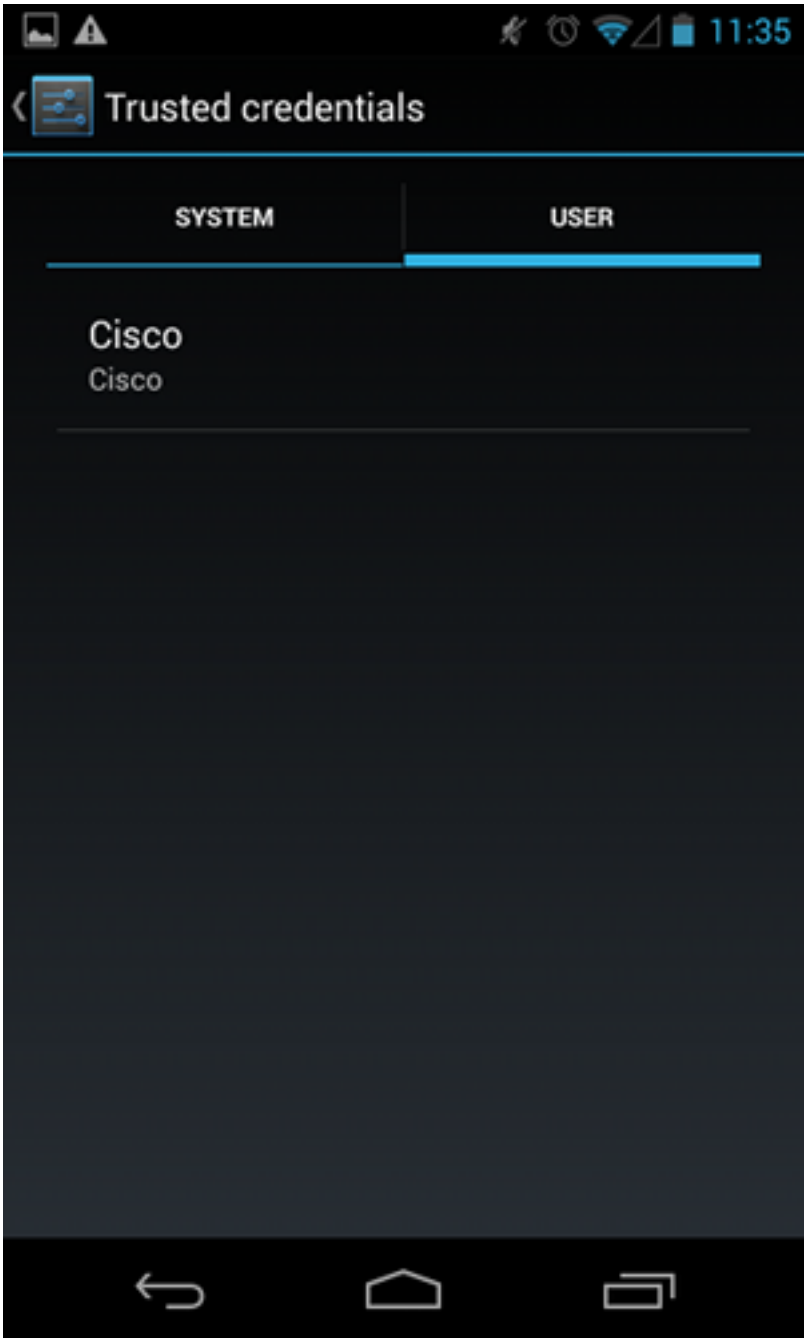
2. Specificare la password utilizzata al momento della generazione del file pfx.



3. Specificare il nome del certificato importato.



4. Per verificare l'installazione del certificato, selezionare **Settings** (Impostazioni) > **Security** (Protezione) > **Trusted Credentials** (Credenziali attendibili). Il nuovo certificato dovrebbe essere visualizzato nell'archivio utenti:



A questo punto, vengono installati un certificato utente e un certificato CA. Il file pfx è un contenitore pkcs12 con il certificato utente e il certificato CA.

Android ha requisiti precisi quando vengono importati i certificati. Ad esempio, per importare correttamente un certificato CA, Android richiede che l'estensione x509v3 Basic Constraint sia impostata su TRUE. Pertanto, quando si genera una CA o si utilizza una CA personalizzata, è importante verificare che abbia l'estensione corretta:

```
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>
```

Autenticazione EAP

Configurazione software Cisco IOS per autenticazione EAP

IKEv2 consente di utilizzare uno stack di protocolli EAP per eseguire l'autenticazione dell'utente. Il gateway VPN si presenta con il certificato. Quando il client considera attendibile il certificato, risponde all'identità della richiesta EAP dal gateway. Il software Cisco IOS utilizza tale identità e invia un messaggio Radius-Request al server di autenticazione, autorizzazione e accounting (AAA). Viene stabilita una sessione EAP-MD5 tra il richiedente (Android) e il server di autenticazione (Access Control Server [ACS] o ISE).

Dopo aver autenticato EAP-MD5, come indicato da un messaggio Radius-Accept, il software Cisco IOS usa la modalità di configurazione per inviare l'indirizzo IP al client e continuare la negoziazione del selettore di traffico.

Si noti che Android ha inviato IKEID=cisco (come configurato). L'IKEID ricevuto sul software Cisco IOS corrisponde a '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
!
```

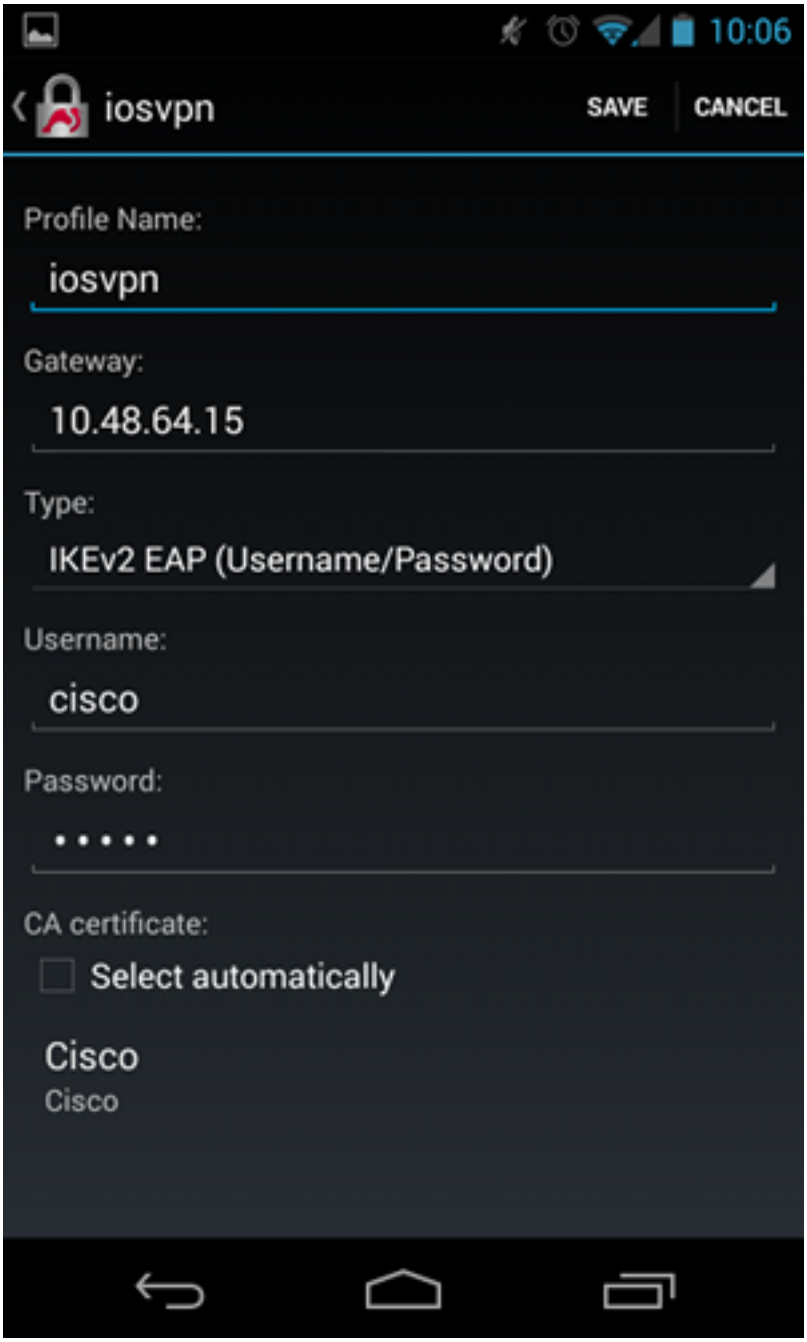
```
crypto ipsec profile PROF
 set transform-set 3DES-MD5
 set ikev2-profile PROF

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
```

Configurazione Android per autenticazione EAP

Android strongSwan deve avere EAP configurato:

1. Disattivare la selezione automatica dei certificati. in caso contrario, nel terzo pacchetto vengono inviati almeno 100 CERT_REQ.

2. Scegliere un certificato (CA) specifico importato nel passaggio precedente; il nome utente e la password devono essere gli stessi del server AAA.



Test di autenticazione EAP

Nel software Cisco IOS, sono i debug più importanti per l'autenticazione EAP. La maggior parte dell'output è stata omessa per chiarezza:

```
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
```

I registri Android indicano:

```
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)
```

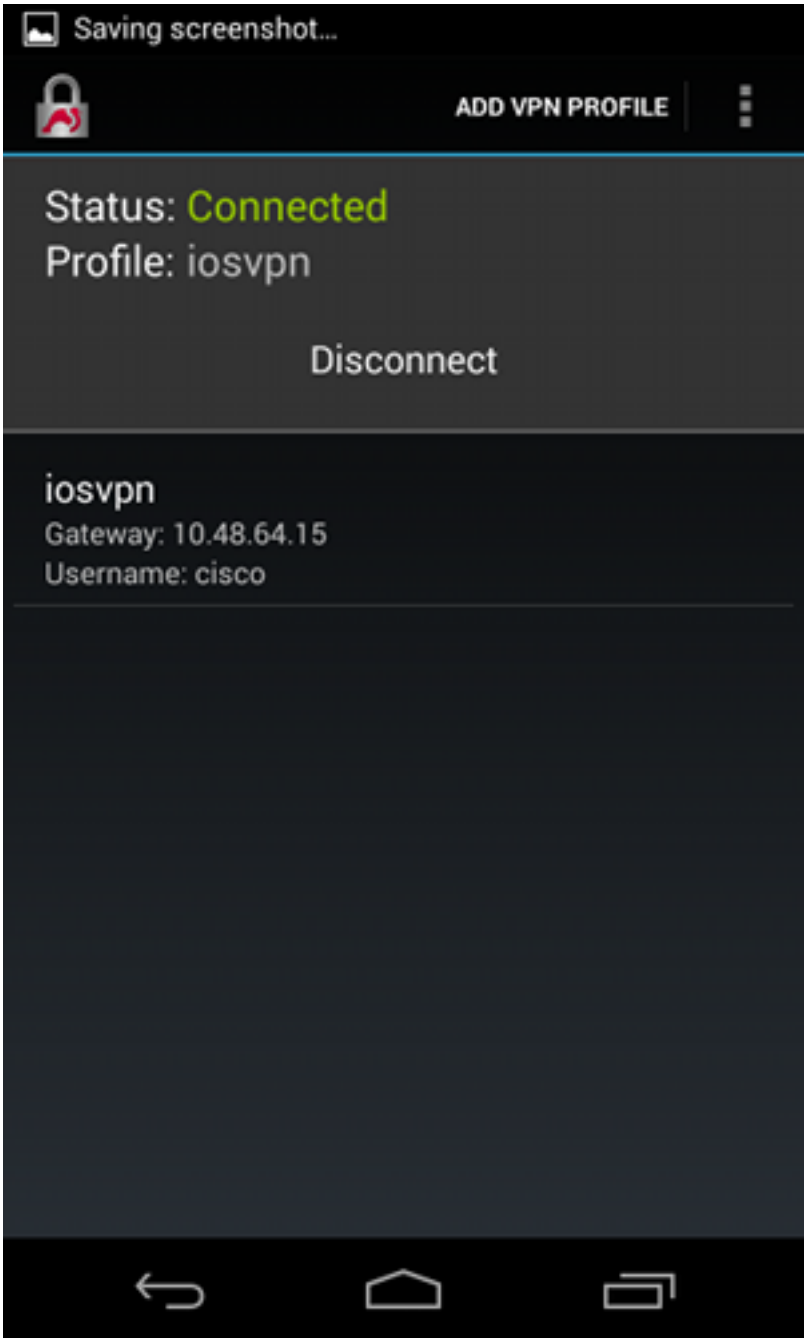
```
09[NET] received packet: from 10.48.64.15[4500] to 10.147.24.153[35564]
(76 bytes)
09[ENC] parsed IKE_AUTH response 2 [ EAP/REQ/TLS ]
09[IKE] server requested EAP_TLS authentication (id 0x59)
09[IKE] EAP method not supported, sending EAP_NAK
09[ENC] generating IKE_AUTH request 3 [ EAP/RES/NAK ]
09[NET] 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
```

Nell'esempio viene mostrato come verificare lo stato sul software 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
```

Queste cifre mostrano come verificare lo stato su Android:


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

Autenticazione RSA

Configurazione software Cisco IOS per autenticazione RSA

Nell'autenticazione Rivest-Shamir-Adleman (RSA), Android invia il certificato per autenticarsi al software Cisco IOS. Ecco perché è necessaria la mappa dei certificati che associa il traffico a un profilo IKEv2 specifico. Autenticazione EAP utente non necessaria.

Questo è un esempio di come è impostata l'autenticazione RSA per un peer remoto:

```
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
```

Configurazione Android per autenticazione RSA

Le credenziali utente sono state sostituite dal certificato utente:

Test di autenticazione RSA

Nel software Cisco IOS, questi sono i debug più importanti per l'autenticazione RSA. La maggior parte dell'output è stata omessa per chiarezza:

```
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
```

I registri Android indicano:

```
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
```

Nel software Cisco IOS, RSA viene utilizzato sia per la firma che per la verifica; nello scenario precedente, per la verifica è stato utilizzato il protocollo EAP:

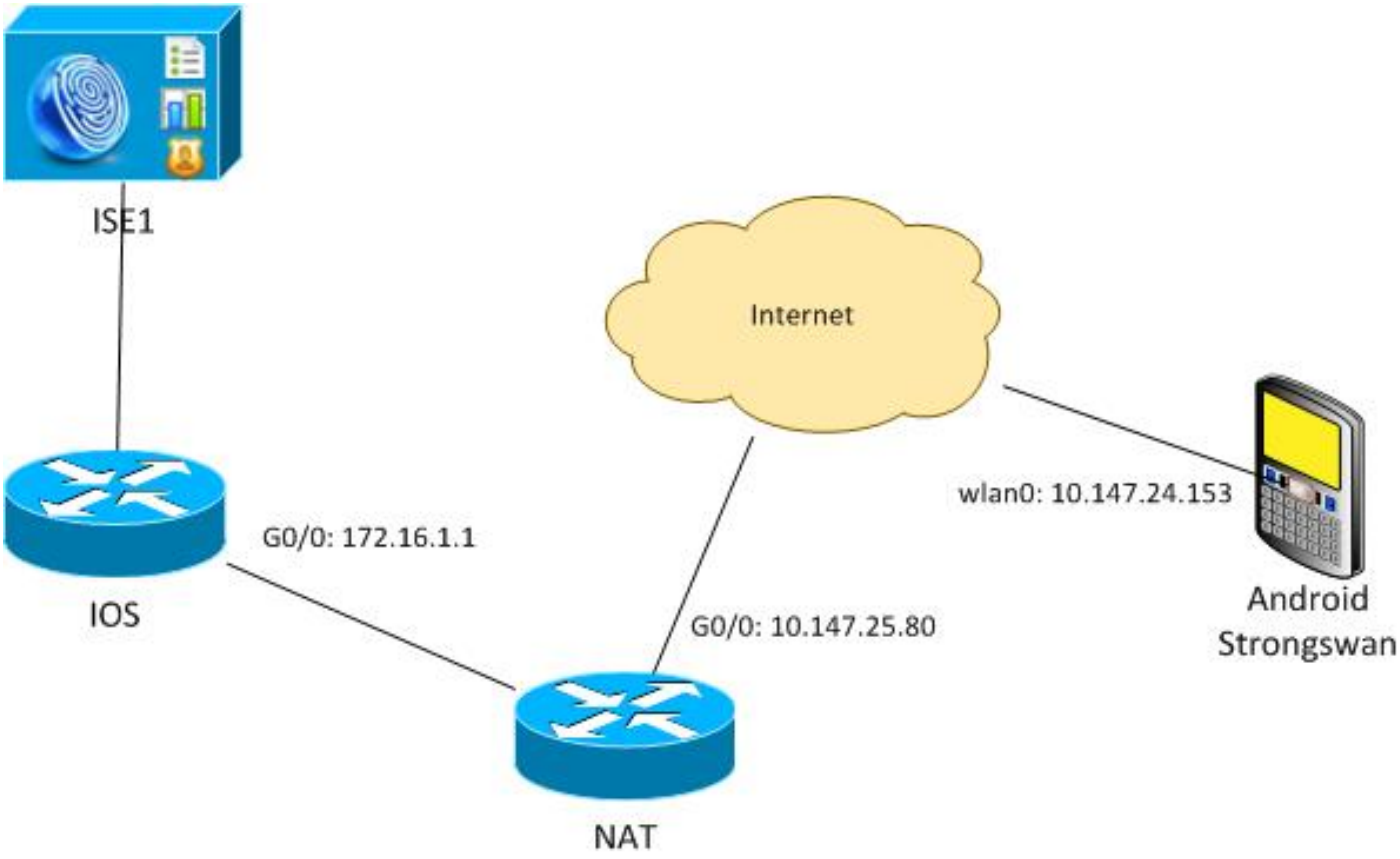
```
BSAN-2900-1#show crypto ikev2 sa detailed
 IPv4 Crypto IKEv2  SA

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 verifica dello stato su Android è simile a quella dello scenario precedente.

VPN Gateway dietro NAT - limitazioni software strongSwan e Cisco IOS

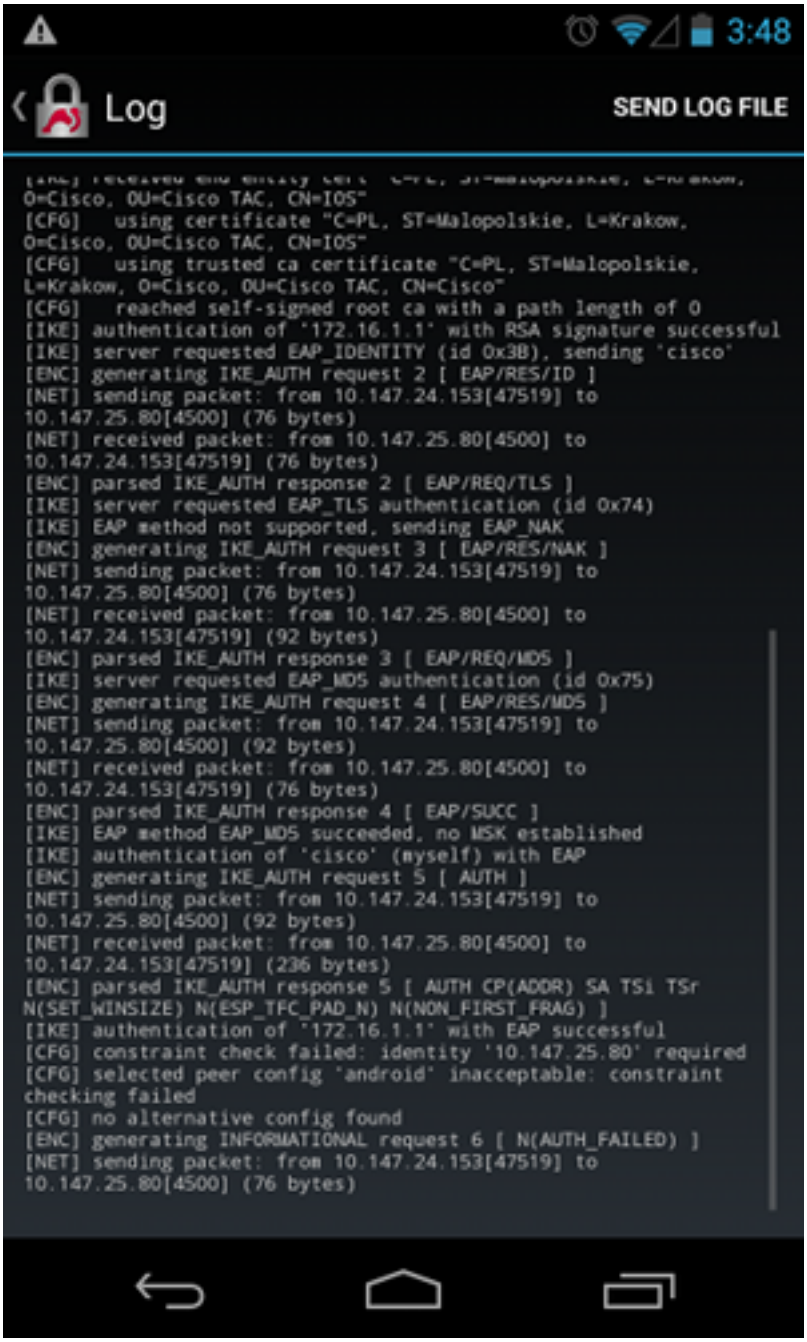
In questo esempio viene illustrata una limitazione delle verifiche dei certificati strongSwan.

Si supponga che l'indirizzo IP del gateway VPN del software Cisco IOS sia convertito in modo statico da 172.16.1.1 a 10.147.25.80. Viene utilizzata l'autenticazione EAP.



Si supponga inoltre che il certificato software Cisco IOS abbia un nome alternativo soggetto sia per 172.16.1.1 che per 10.147.25.80.

Dopo l'autenticazione EAP, Android esegue la verifica e tenta di trovare l'indirizzo IP del peer utilizzato nella configurazione Android (10.147.25.80) nell'estensione del nome alternativo del soggetto. La verifica non riesce:

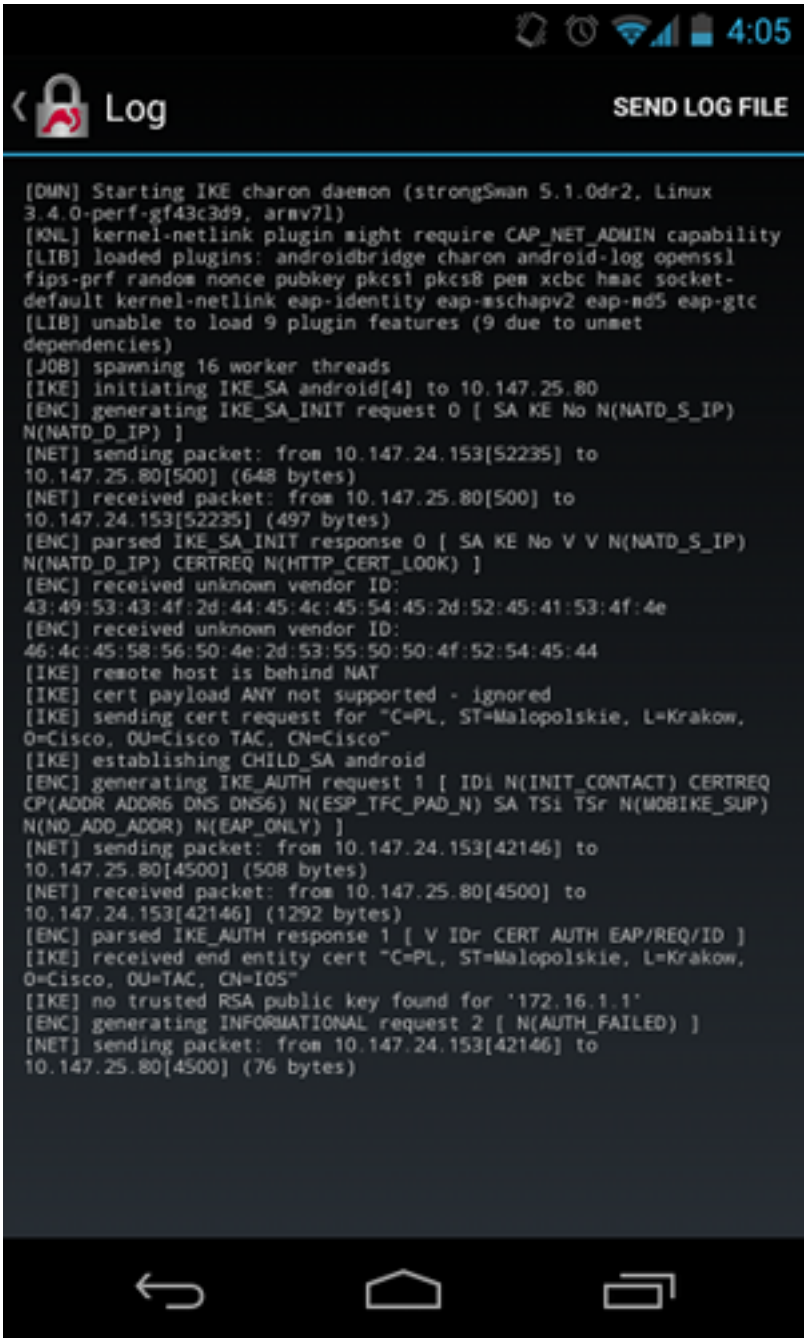


I registri indicano:

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

L'errore si è verificato perché Android è in grado di leggere solo la prima estensione del nome alternativo del soggetto (172.16.1.1).

Si supponga ora che il certificato software Cisco IOS abbia entrambi gli indirizzi nel campo Nome alternativo soggetto ma nell'ordine inverso: 10.147.25.80 e 172.16.1.1. Android esegue la convalida quando riceve l'IKEID, ovvero l'indirizzo IP del gateway VPN (172.16.1.1), nel terzo pacchetto:



Nel registro verranno visualizzati:

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

Pertanto, quando Android riceve l'IKEID, deve trovarlo nel Nome alternativo soggetto e può utilizzare solo il primo indirizzo IP.

> **Nota:** Nell'autenticazione EAP, l'IKEID inviato dal software Cisco IOS è l'indirizzo IP per impostazione predefinita. Nell'autenticazione RSA, IKEID è il DN del certificato per impostazione predefinita. Per modificare manualmente questi valori, usare il comando **identity** nel profilo ikev2.

Verifica

Le procedure di verifica e test sono disponibili negli esempi di configurazione.

Risoluzione dei problemi

Le informazioni contenute in questa sezione permettono di risolvere i problemi relativi alla configurazione.

CA strongSwan multipla CERT_REQ

Quando l'impostazione del certificato su strongSwan è Selezione automatica (impostazione predefinita), Android invia CERT_REQ per tutti i certificati attendibili nell'archivio locale nel terzo pacchetto. Il software Cisco IOS potrebbe rifiutare la richiesta perché riconosce un numero elevato di richieste di certificati come attacco Denial of Service:

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

Origine tunnel su DVTI

Sebbene sia abbastanza comune impostare l'origine del tunnel su un'interfaccia VTI (Virtual Tunnel Interface), non è necessario in questo caso. Si supponga che il comando **tunnel source** si trovi in un VTI (DVTI) dinamico:

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

Dopo l'autenticazione, se il software Cisco IOS tenta di creare un'interfaccia di accesso virtuale clonata da un modello virtuale, restituisce un errore:

```
*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
```

Due secondi dopo l'errore, il software Cisco IOS riceve una nuova trasmissione IKE_AUTH da Android. Quel pacchetto è scartato.

Bug e richieste di miglioramenti del software Cisco IOS

- Cisco Bug ID CSCui46418, "IOS Ikev2 ip address sent as identity for RSA authentication" (Indirizzo IP IOS Ikev2 inviato come identità per l'autenticazione RSA).
  Questo bug non è un problema, purché strongSwan possa vedere un nome alternativo del soggetto (l'indirizzo IP) corretto quando cerca l'IKEID nel certificato per eseguire la verifica.
- Cisco Bug ID CSCui44976, "IOS PKI ha visualizzato in modo errato il nome alternativo del

soggetto dell'estensione X509v3."
Questo bug si verifica solo quando il nome alternativo del soggetto contiene più indirizzi IP. Verrà visualizzato solo l'ultimo indirizzo IP, ma ciò non influirà sull'utilizzo del certificato. L'intero certificato viene inviato ed elaborato correttamente.

- Cisco Bug ID CSCui44783, "IOS ENH PKI ability to generate CSR with subject-alt-name extension".
- Cisco Bug ID CSCui44335, "ASA ENH Certificate x509 extensions displayed" (Visualizzate le estensioni del certificato ENH ASA x509).

Informazioni correlate

- Guida alla configurazione della VPN di Cisco IOS 15.3
- Guida di riferimento ai comandi di Cisco IOS 15.3
- Guida alla configurazione di Cisco IOS Flex VPN
- Documentazione e supporto tecnico – Cisco Systems