IKEv2 de Android strongSwan para Cisco IOS com autenticação EAP e RSA

Contents

Introduction

Este documento descreve como configurar a versão móvel do strongSwan para acessar um gateway de VPN do software Cisco IOS® através do protocolo IKEv2 (Internet Key Exchange Version 2).

Três exemplos são apresentados:

- Telefone Android com strongSwan que se conecta ao gateway VPN do software Cisco IOS com autenticação Extensible Authentication Protocol - Message Digest 5 (EAP-MD5).
- Telefone Android com strongSwan que se conecta ao gateway de VPN do software Cisco

IOS com autenticação de certificado (RSA).
- Telefone Android com strongSwan que se conecta ao gateway VPN do software Cisco IOS por trás da Network Address Translation (NAT). Há um requisito para ter dois ramais x509 Nome alternativo do assunto no certificado do gateway VPN.

As limitações do software Cisco IOS e do strongSwan também estão incluídas.

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Conhecimento básico da configuração do OpenSSL
- Conhecimento básico da configuração da interface de linha de comando (CLI) do software Cisco IOS
- Conhecimento básico de IKEv2

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

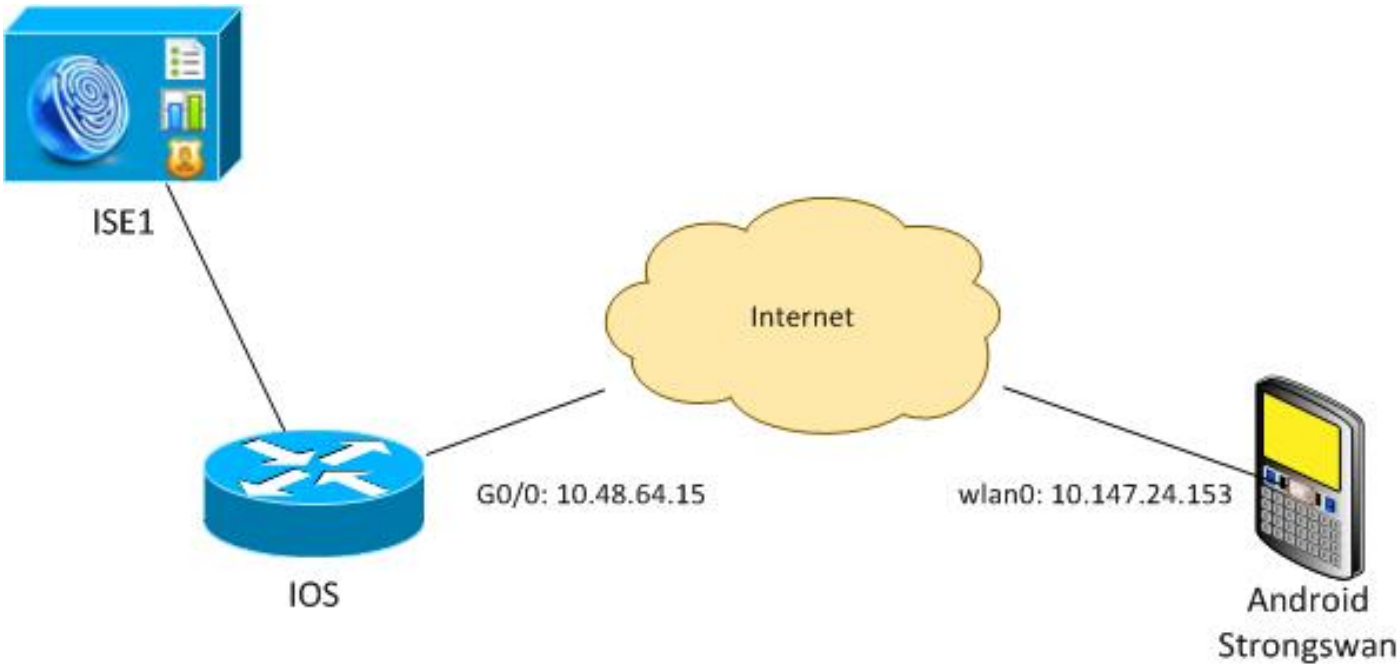
- Android 4.0 ou posterior com strongSwan
- Software Cisco IOS versão 15.3T ou posterior
- Software Cisco Identity Services Engine (ISE), versão 1.1.4 e posterior

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.

Configurar

> **Notas:**
>
> A ferramenta Output Interpreter (exclusiva para clientes registrados) é compatível com alguns comandos de exibição.. Use a ferramenta Output Interpreter para visualizar uma análise do resultado gerado pelo comando show..
>
> Consulte Informações Importantes sobre Comandos de Depuração antes de usar comandos **debug.**

Diagrama de Rede



O Android strongSwan estabelece um túnel IKEv2 com um gateway do software Cisco IOS para acessar redes internas com segurança.

Inscrição de certificado

Os certificados são um pré-requisito para a autenticação baseada em EAP e RSA.

No cenário de autenticação EAP, um certificado é necessário somente no gateway VPN. O cliente se conecta ao software Cisco IOS somente quando o software apresenta um certificado assinado por uma autoridade de certificação (CA) confiável no Android. Uma sessão EAP é então iniciada para que o cliente se autentique no software Cisco IOS.

Para autenticação baseada em RSA, ambos os endpoints devem ter um certificado correto.

Quando um endereço IP é usado como um peer-ID, há requisitos adicionais para o certificado. O Android strongSwan verifica se o endereço IP do gateway VPN está incluído no nome alternativo do assunto da extensão x509. Caso contrário, o Android descarta a conexão; essa é uma boa prática, bem como uma recomendação do RFC 6125.

O OpenSSL é usado como uma CA porque o software Cisco IOS tem uma limitação: ele não pode gerar certificados com uma extensão que inclui um endereço IP. Todos os certificados são gerados pelo OpenSSL e importados para o Android e o software Cisco IOS.

No software Cisco IOS, o comando **subject-alt-name** pode ser usado para criar uma extensão que inclua um endereço IP, mas o comando funciona somente com certificados autoassinados. ID de bug Cisco CSCui44783, "IOS ENH PKI ability to generate CSR with subject-alt-name extension" é uma solicitação de aprimoramento para permitir que o software Cisco IOS gere a extensão para todos os tipos de inscrição.

Este é um exemplo dos comandos que geram uma 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 é um arquivo de configuração. O ramal da CA deve ser definido como 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
```

Os comandos que geram um certificado são muito semelhantes para o software Cisco IOS e Android. Este exemplo pressupõe que já existe uma CA usada para assinar o certificado:

```
#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 é um arquivo de configuração. A extensão Nome do assunto alternativo é uma configuração chave. Neste exemplo, o ramal da CA está definido como 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
```

Um certificado deve ser gerado para o software Cisco IOS e para o Android.

O endereço IP 10.48.64.15 pertence ao gateway do software Cisco IOS. Ao gerar um certificado para o software Cisco IOS, certifique-se de que o subjectAltName esteja definido como 10.48.64.15. O Android valida o certificado recebido do software Cisco IOS e tenta encontrar seu endereço IP no subjectAltName.

Cisco IOS Software

O software Cisco IOS precisa ter um certificado correto instalado para autenticação baseada em RSA e EAP.

O arquivo pfx (que é um contêiner pkcs12) para o software Cisco IOS pode ser importado:

```
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.
```

Use o comando **show crypto pki certificate verbose** para verificar se a importação foi bem-sucedida:

```
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

Para autenticação baseada em EAP, Andorid precisa ter apenas o certificado de CA correto instalado.

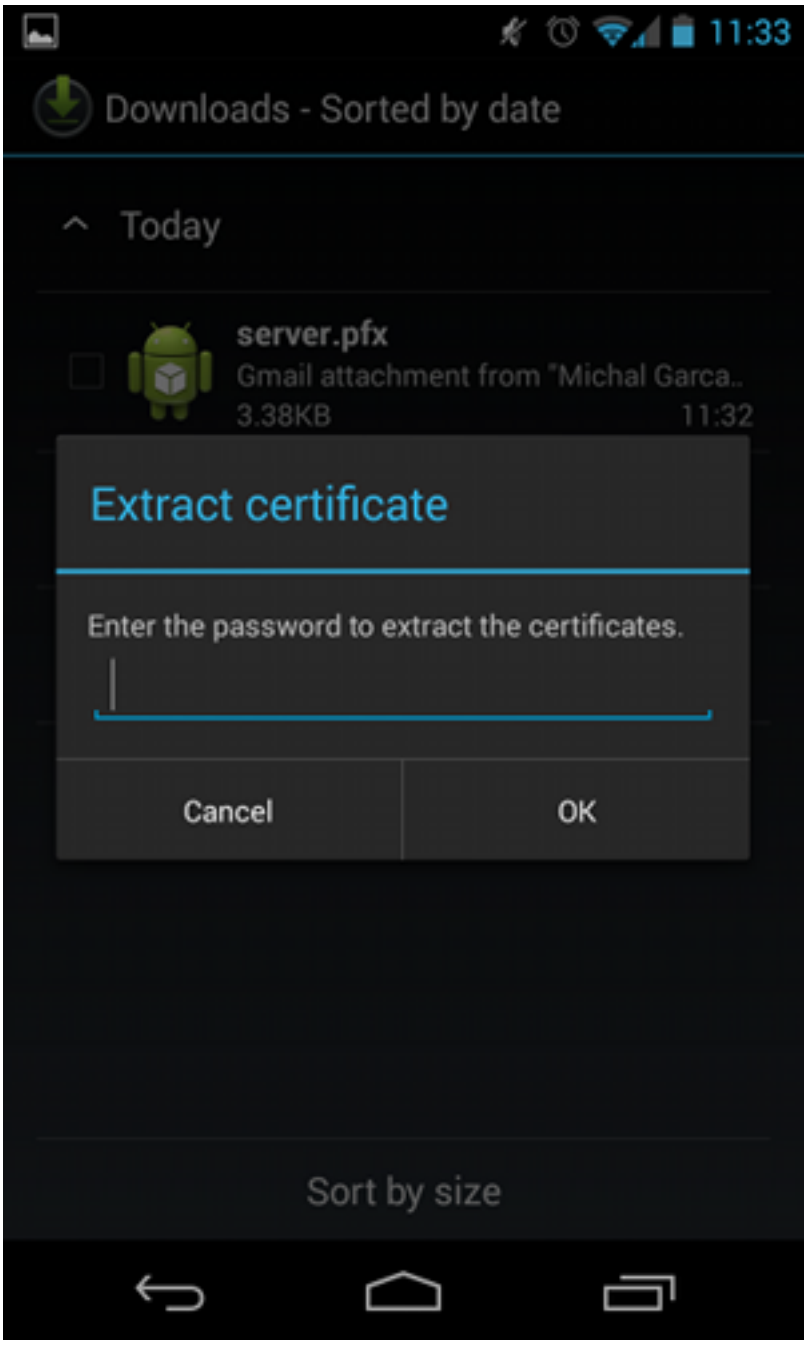
Para autenticação baseada em RSA, Andorid precisa ter o certificado CA e seu próprio certificado

instalados.

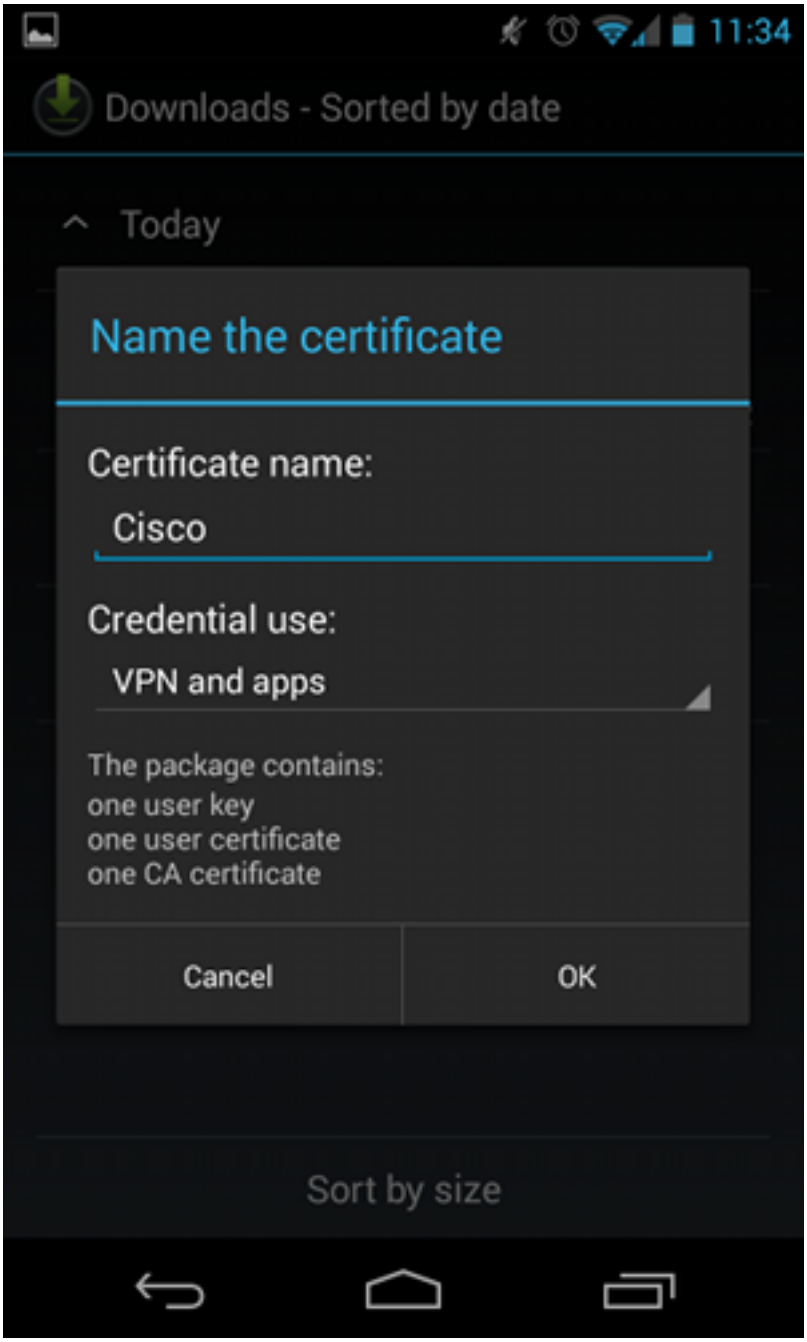
Este procedimento descreve como instalar os dois certificados:

1. Envie o arquivo pfx por e-mail e abra-o.

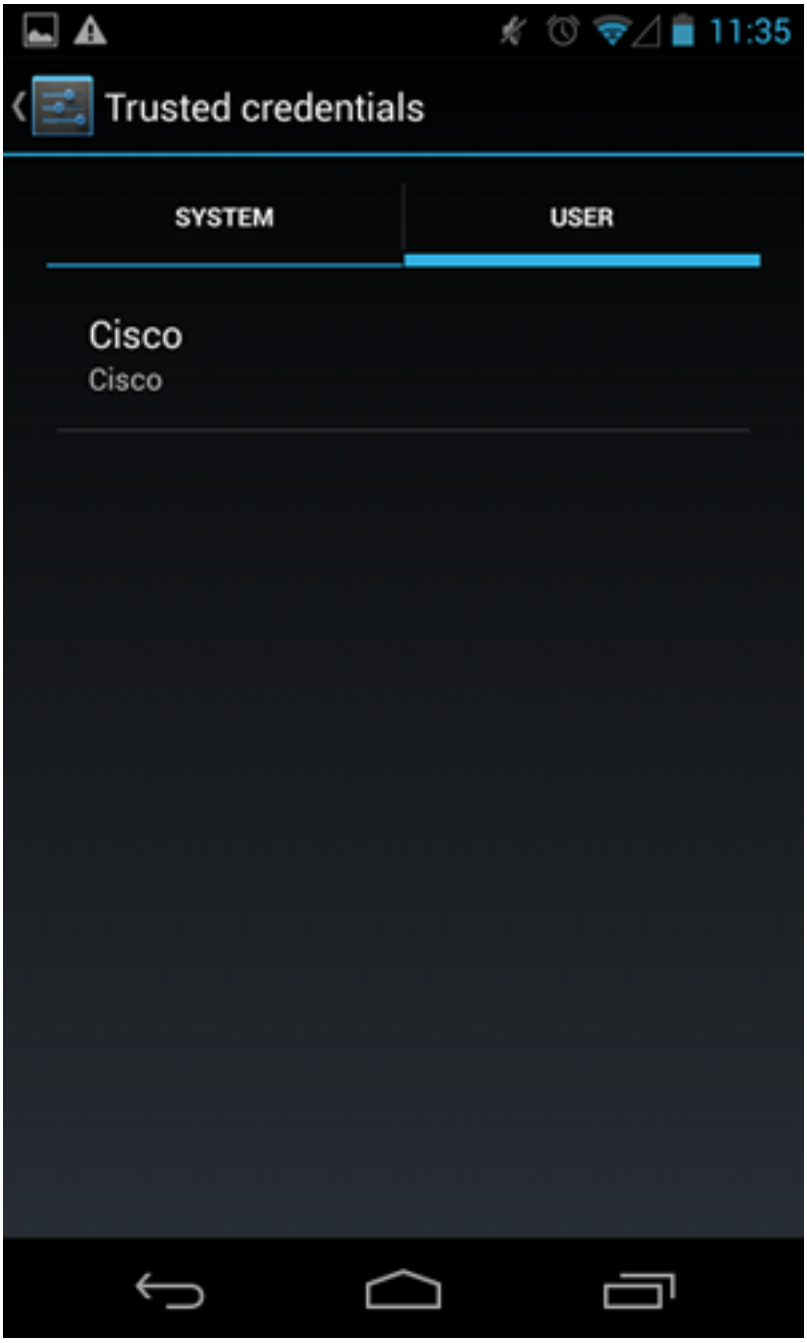
2. Forneça a senha que foi usada quando o arquivo pfx foi gerado.



3. Forneça o nome do certificado importado.



4. Navegue até **Configurações** > **Segurança** > **Credenciais confiáveis** para verificar a instalação do certificado. O novo certificado deve aparecer no repositório de usuários:



Neste ponto, um certificado de usuário e um certificado CA são instalados. O arquivo pfx é um contêiner pkcs12 com o certificado do usuário e o certificado CA.

O Android tem requisitos precisos quando os certificados são importados. Por exemplo, para que um certificado CA seja importado com êxito, o Android exige que a CA de restrição básica da extensão x509v3 seja definida como TRUE. Assim, quando você gera uma CA ou usa sua própria CA, é importante verificar se ela tem o ramal correto:

```
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>
```

Autenticação EAP

Configuração do Cisco IOS Software para Autenticação EAP

O IKEv2 permite o uso de uma pilha de protocolos EAP para executar a autenticação do usuário. O gateway VPN apresenta o certificado. Quando o cliente confia nesse certificado, ele responde à identidade de solicitação EAP do gateway. O software Cisco IOS usa essa identidade e envia uma mensagem Radius-Request para o servidor de autenticação, autorização e contabilização (AAA), e uma sessão EAP-MD5 é estabelecida entre o requerente (Android) e o servidor de autenticação (Access Control Server [ACS] ou ISE).

Após a autenticação EAP-MD5 bem-sucedida, conforme indicado por uma mensagem de RADIUS-Accept, o software Cisco IOS usa o modo de configuração para enviar o endereço IP ao cliente e continuar com a negociação do seletor de tráfego.

Observe que o Android enviou IKEID=cisco (conforme configurado). Este IKEID recebido no software Cisco IOS corresponde ao '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
!
```

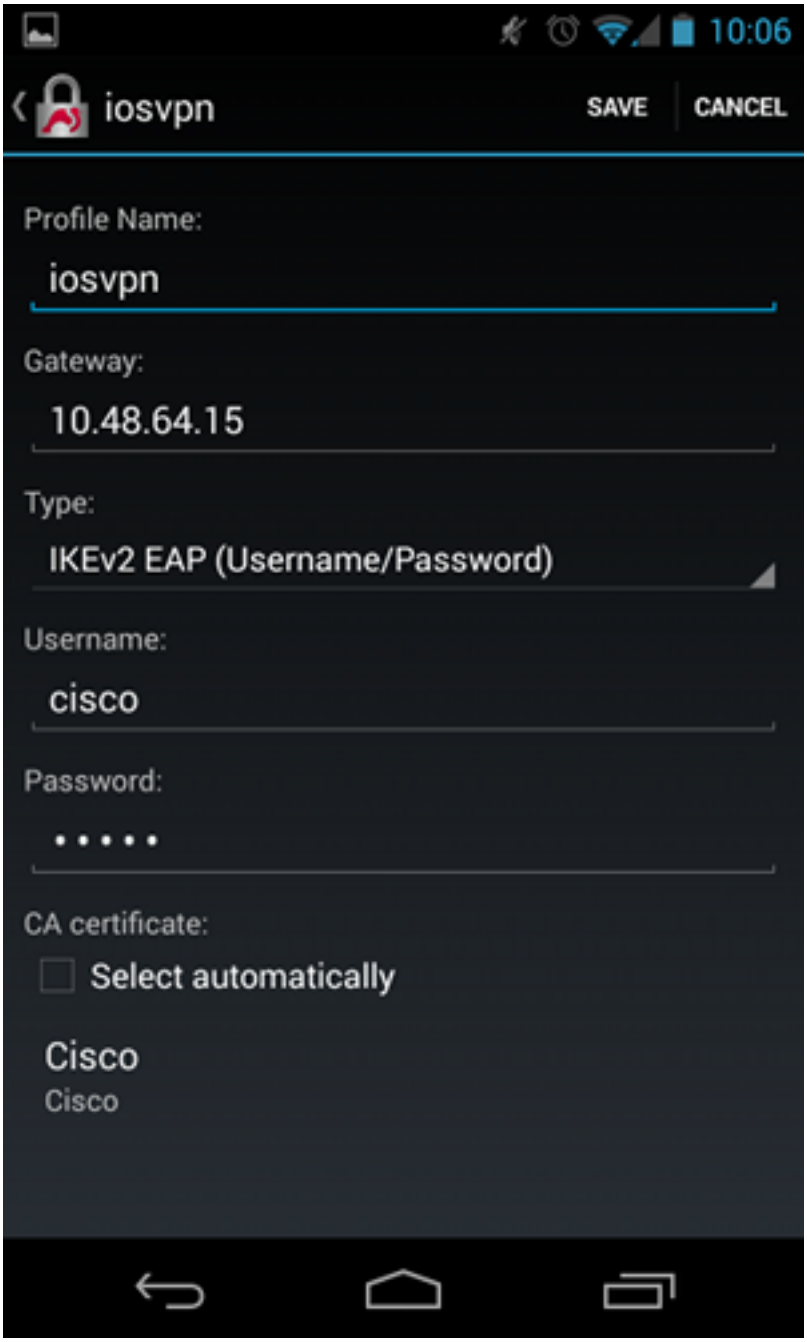
```
crypto ipsec profile PROF
 set transform-set 3DES-MD5
 set ikev2-profile PROF

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
```

Configuração do Android para Autenticação EAP

O Android strongSwan deve ter o EAP configurado:

1. Desativar seleção automática de certificado; caso contrário, 100 ou mais CERT_REQs são enviados no terceiro pacote.

2. Escolha um certificado específico (CA) que foi importado na etapa anterior; o nome de usuário e a senha devem ser iguais aos do servidor AAA.



Teste de autenticação EAP

No software Cisco IOS, essas são as depurações mais importantes para autenticação EAP. A maior parte da saída foi omitida para maior clareza:

```
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
```

Os registros do Android indicam:

```
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)
```

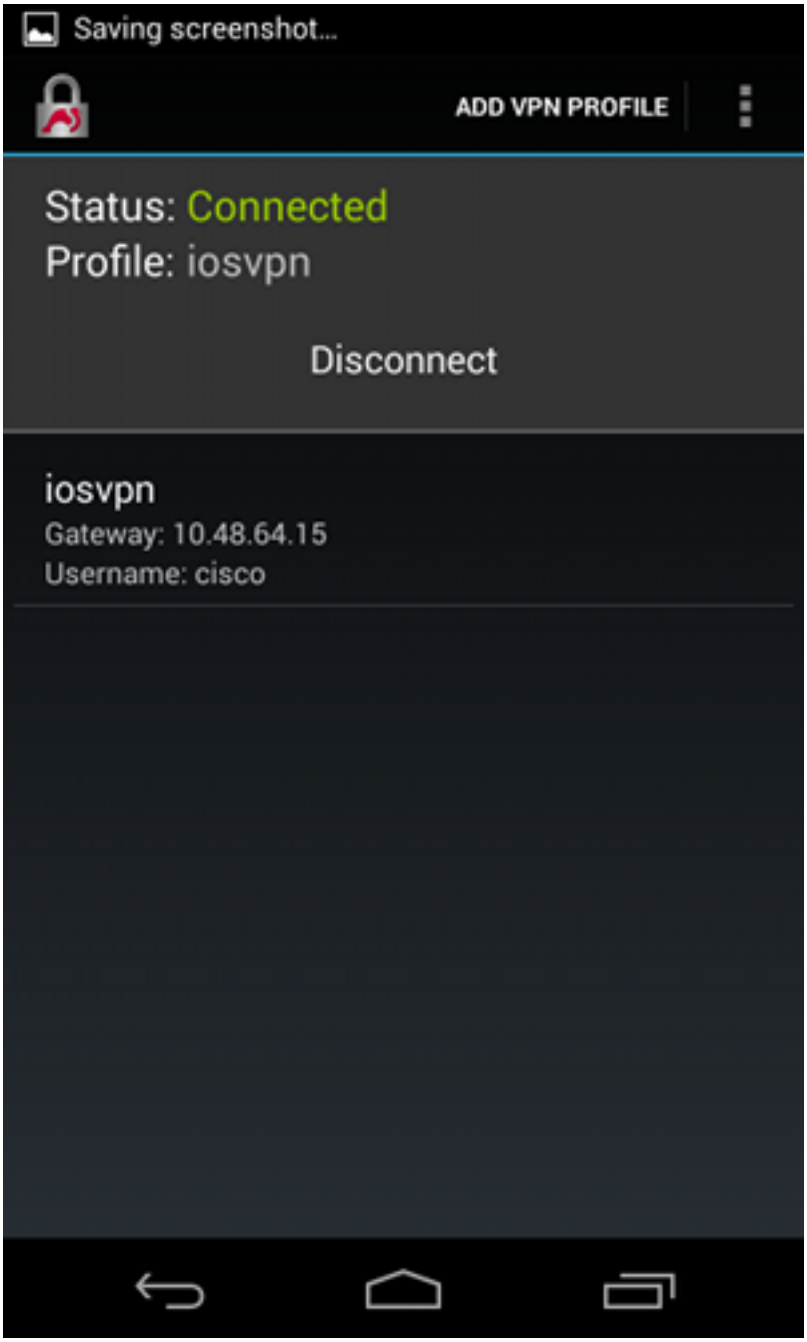
```
09[NET] received packet: from 10.48.64.15[4500] to 10.147.24.153[35564]
(76 bytes)
09[ENC] parsed IKE_AUTH response 2 [ EAP/REQ/TLS ]
09[IKE] server requested EAP_TLS authentication (id 0x59)
09[IKE] EAP method not supported, sending EAP_NAK
09[ENC] generating IKE_AUTH request 3 [ EAP/RES/NAK ]
09[NET]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

Este exemplo mostra como verificar o status no 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
```

Estas figuras mostram como verificar o status no Android:


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

Autenticação RSA

Configuração do software Cisco IOS para autenticação RSA

Na autenticação RSA (Rivest-Shamir-Adleman), o Android envia o certificado para autenticação no software Cisco IOS. É por isso que o mapa de certificados que vincula esse tráfego a um perfil IKEv2 específico é necessário. A autenticação EAP do usuário não é necessária.

Este é um exemplo de como a autenticação RSA para um peer remoto é definida:

```
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
```

Configuração do Android para Autenticação RSA

As credenciais de usuário foram substituídas pelo certificado de usuário:

Teste de autenticação RSA

No software Cisco IOS, essas são as depurações mais importantes para autenticação RSA. A maior parte da saída foi omitida para maior clareza:

```
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
```

Os registros do Android indicam:

```
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
```

No software Cisco IOS, o RSA é usado para assinatura e verificação; no cenário anterior, o EAP foi utilizado para a verificação:

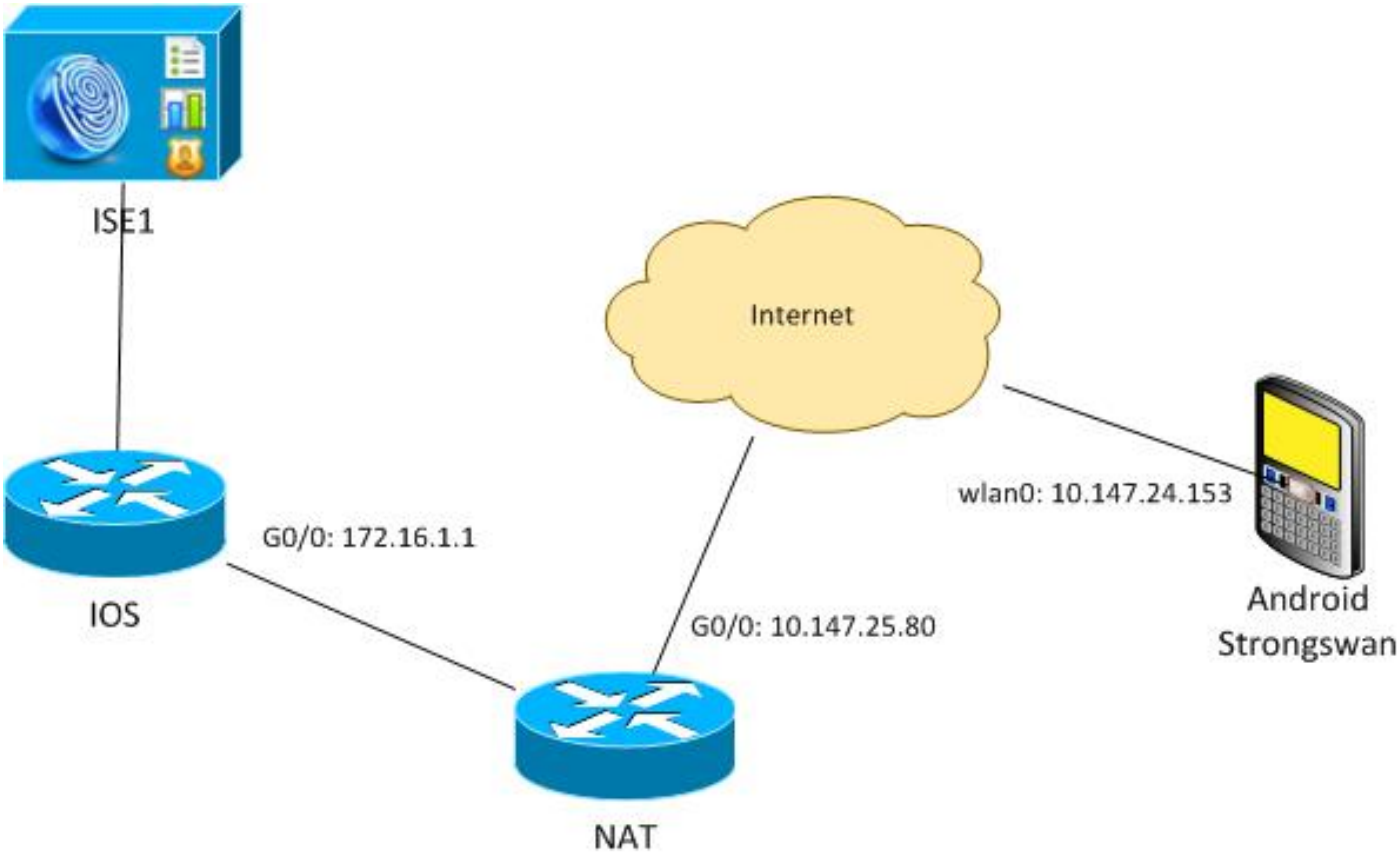
```
BSAN-2900-1#show crypto ikev2 sa detailed
 IPv4 Crypto IKEv2  SA

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
```

A verificação de status no Android é semelhante à do cenário anterior.

Gateway VPN atrás de NAT - strongSwan e limitações do software Cisco IOS

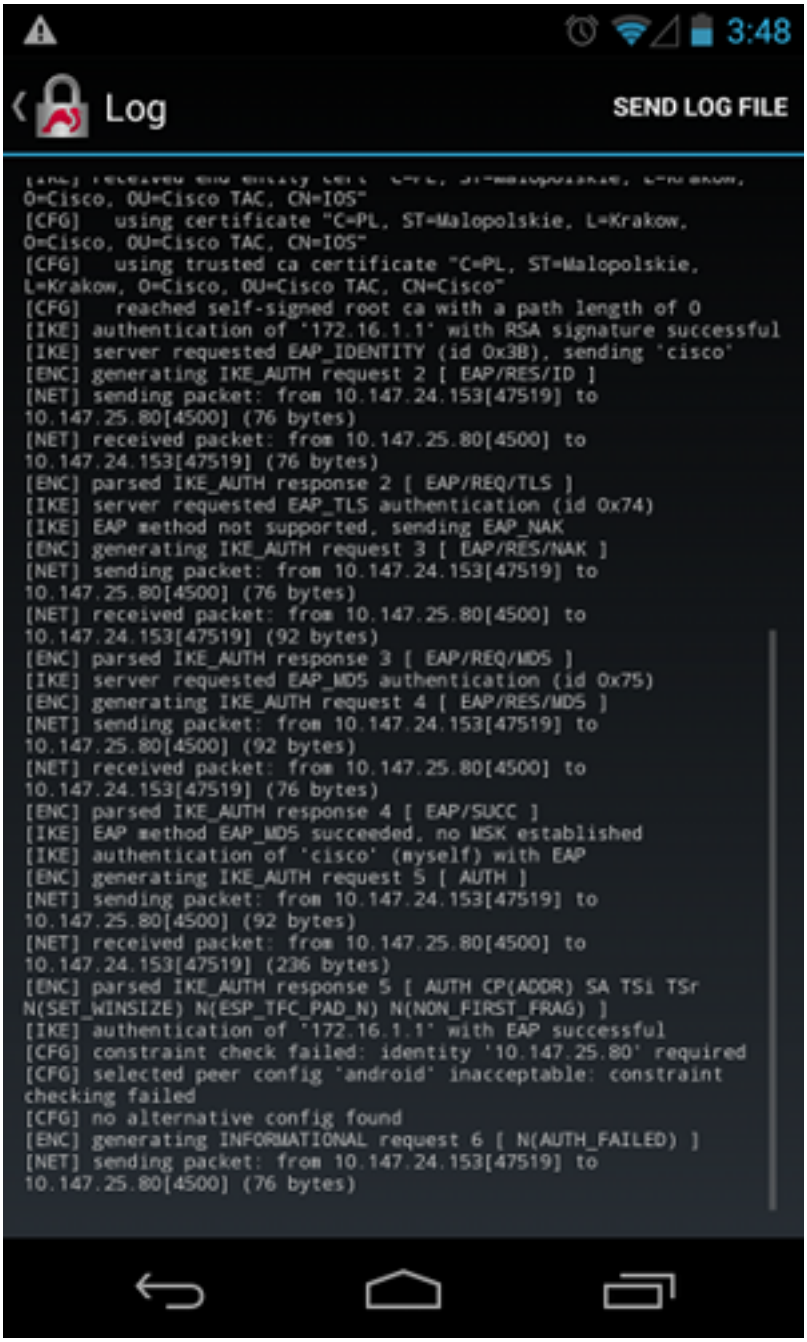
Este exemplo explica uma limitação das verificações do certificado strongSwan.

Suponha que o endereço IP do gateway de VPN do software Cisco IOS seja convertido estaticamente de 172.16.1.1 para 10.147.25.80. A autenticação EAP é usada.



Suponha também que o certificado do software Cisco IOS tenha um nome alternativo de assunto para 172.16.1.1 e 10.147.25.80.

Após a autenticação EAP bem-sucedida, o Android executa a verificação e tenta encontrar o endereço IP do peer que foi usado na configuração do Android (10.147.25.80) na extensão do nome alternativo do assunto. Falha na verificação:

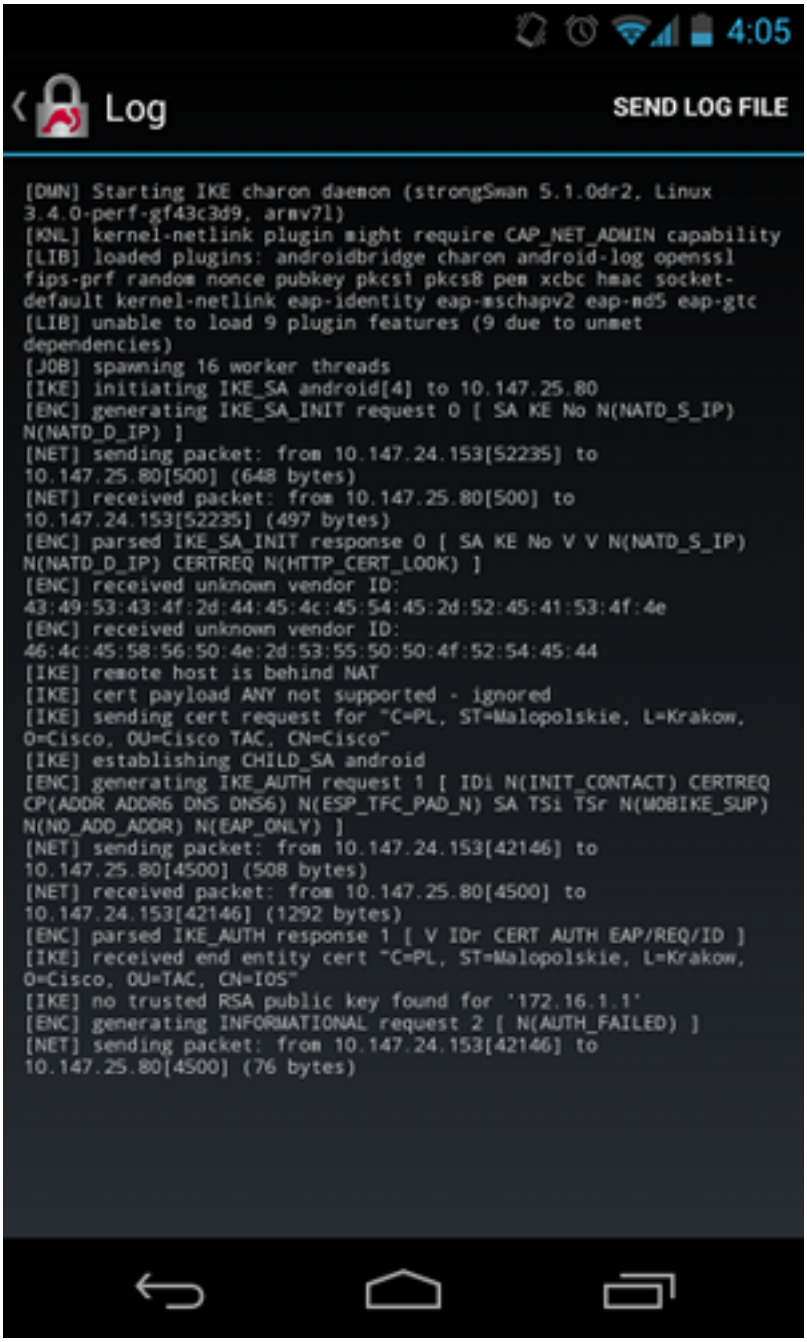


Os registros indicam:

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

A falha ocorreu porque o Android pode ler somente a primeira extensão do nome alternativo do assunto (172.16.1.1).

Agora, suponha que o certificado do software Cisco IOS tenha ambos os endereços em Nome alternativo do assunto, mas na ordem inversa: 10.147.25.80 e 172.16.1.1. O Android executa a validação quando recebe o IKEID, que é o endereço IP do gateway VPN (172.16.1.1), no terceiro pacote:



Agora o registro mostra:

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

Assim, quando o Android recebe o IKEID, ele precisa encontrar o IKEID no nome alternativo do assunto e pode usar apenas o primeiro endereço IP.

> **Note**: Na autenticação EAP, o IKEID enviado pelo software Cisco IOS é o endereço IP por padrão. Na autenticação RSA, o IKEID é o DN do certificado por padrão. Use o comando **identity** no perfil ikev2 para alterar esses valores manualmente.

Verificar

Os procedimentos de verificação e teste estão disponíveis nos exemplos de configuração.

Troubleshoot

Esta seção fornece informações que podem ser usadas para o troubleshooting da sua configuração.

strongSwan CA Múltipla CERT_REQ

Quando a configuração do certificado em strongSwan é Seleção automática (o padrão), o Android envia CERT_REQ para todos os certificados confiáveis na loja local no terceiro pacote . O software Cisco IOS pode descartar a solicitação porque reconhece um grande número de solicitações de certificado como um ataque de negação de serviço:

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

Origem do túnel em DVTI

Embora seja bastante comum definir a origem do túnel em uma interface de túnel virtual (VTI), não é necessário aqui. Suponha que o comando **tunnel source** está em um VTI dinâmico (DVTI):

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

Após a autenticação, se o software Cisco IOS tentar criar uma interface de acesso virtual clonada a partir de um modelo virtual, ele retornará um erro:

```
*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
```

Dois segundos após a falha, o software Cisco IOS recebe um IKE_AUTH retransmitido do Android. Esse pacote é descartado.

Bugs do software Cisco IOS e solicitações de aprimoramento

- ID de bug Cisco CSCui46418, "Endereço IP Ikev2 do IOS enviado como identidade para autenticação RSA".
  Este bug não é um problema, desde que strongSwan possa ver um nome alternativo de assunto correto (o endereço IP) quando procurar o IKEID no certificado para executar a verificação.
- ID de bug Cisco CSCui4976, "PKI IOS incorretamente exibido o nome alternativo do assunto da extensão X509v3."

Esse bug ocorre somente quando há vários endereços IP no nome alternativo do assunto. Somente o último endereço IP é exibido, mas isso não afeta o uso do certificado. O certificado inteiro é enviado e processado corretamente.

- ID de bug Cisco CSCui44783, "IOS ENH PKI habilidade para gerar CSR com extensão de nome alternativo de assunto".
- ID de bug Cisco CSCui44335, "Extensões do certificado ASA ENH x509 exibidas."

Informações Relacionadas

- Guia de configuração de VPN do Cisco IOS 15.3
- Referência de comando do Cisco IOS 15.3
- Guia de configuração do Cisco IOS Flex VPN
- Suporte Técnico e Documentação - Cisco Systems