

# Pesquisa defeitos edições comuns GETVPN

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de fundo - Ferramentas de Troubleshooting GETVPN](#)

[Controle ferramentas para debug planas](#)

[comandos show](#)

[Syslogs](#)

[Domínio do grupo do traço do evento da interpretação \(GDOI\)](#)

[GDOI condicional debuga](#)

[Cripto e GDOI globais debugam](#)

[Ferramentas para debug planas dos dados](#)

[Troubleshooting](#)

[Preparação da facilidade de registro e outros melhores prática](#)

[Pesquisa defeitos o estabelecimento de IKE](#)

[Pesquisa defeitos o registro inicial](#)

[Pesquisa defeitos edições Política-relacionadas](#)

[O problema de política ocorre antes do registro \(a política relativa do Falha-fim\)](#)

[O problema de política ocorre registro do CARGO, e refere-se a política global que é empurrada](#)

[O problema de política ocorre registro do CARGO, e refere-se a fusão da política global e o Local cancela](#)

[Pesquisa defeitos Rekey edições](#)

[Pesquisa defeitos a Anti-repetição com base no período \(TBAR\)](#)

[Pesquisa defeitos a Redundância KS](#)

[FAQ](#)

[Pode um roteador configurado como KS para um grupo GETVPN igualmente para funcionar como um GM para o mesmos grupo?](#)

[Informações Relacionadas](#)

## Introdução

Este documento descreve o que debuga para recolher para a maioria grupo comum das edições cifradas do transporte VPN (GETVPN).

# Pré-requisitos

## Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- GETVPN
- Uso do servidor de SYSLOG

## Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

## Informações de fundo - Ferramentas de Troubleshooting GETVPN

GETVPN fornece um grupo extensivo de ferramentas de Troubleshooting a fim facilitar o processo da pesquisa de defeitos. É importante compreender quais destas ferramentas estão disponíveis, e quando são apropriados para cada tarefas de Troubleshooting. Ao pesquisar defeitos, é sempre uma boa ideia começar com menos métodos intrusivos, de modo que o ambiente de produção não seja impactado negativamente. A fim ajudar a esse processo, esta seção descreve algumas das ferramentas de uso geral disponíveis:

### Controle ferramentas para debug planas

#### Comandos show

Os comandos show são de uso geral a fim mostrar operações do tempo de execução em um

ambiente GETVPN.

## Syslogs

GETVPN tem um grupo aumentado de mensagens do syslog para eventos e condições de erro significativos do protocolo. Este deve sempre ser o primeiro lugar a olhar antes que você execute alguns debugar.

## Domínio do grupo do traço do evento da interpretação (GDOI)

Esta característica foi adicionada na versão 15.1(3)T. O rastreamento de evento oferece o peso leve, sempre-no seguimento para eventos significativos e erros GDOI. Há igualmente saída-PATH que segue com o retorno de monitoramento permitido para condições de exceção.

## GDOI condicional debuga

Esta característica foi adicionada na versão 15.1(3)T. Reserva filtrado debuga para um dispositivo dado baseado no endereço de peer, e deve sempre ser usada quando possível, especialmente no server chave.

## Cripto e GDOI globais debugam

Estes são todos os vários GETVPM debugam. Admins deve usar o cuidado ao debugar em ambientes em grande escala. Com GDOI debuga, cinco níveis de debug são fornecidos para uma granularidade mais adicional da eliminação de erros:

```
GM1#debug crypto gdoi gm rekey ?
all-levels All levels
detail Detail level
error Error level
event Event level
packet Packet level
terse Terse level
```

<b>Nível de debug</b>	<b>O que você obterá</b>
Erro	Condições de erro
Sóbrio	Mensagens importantes ao usuário e às edições do protocolo

Evento	As transições de estado e os eventos como enviam e recebem rekeys
Detalhe	A maioria detalhados debugam a informação de mensagem
Pacote	Inclui a descarga de informação do pacote detalhada
Todos	Todo o acima

## Ferramentas para debug planas dos dados

Estão aqui as ferramentas para debug do plano de alguns dados:

- Listas de acesso
- Explicar da Precedência IP
- Netflow
- Contadores de interface
- Contadores criptos
- Cisco Express Forwarding (CEF) IP global e contadores de queda da Por-característica
- Captura de pacote de informação encaixada (EPC)
- O plano dos dados debuga (o pacote IP e o CEF debugam)

## Troubleshooting

### Preparação da facilidade de registro e outros melhores prática

Antes que você comece a pesquisar defeitos, assegure-se de que você prepare a facilidade de registro como descrita aqui. Alguns melhores prática são alistados igualmente aqui:

- Verifique a quantidade de memória livre do roteador, e configurar o **logging buffered debugging a um grande valor** (10 MB ou mais se possível).
- Desabilite o registro ao console, ao monitor, e aos servidores de SYSLOG.
- Recupere o índice do logging buffer com o **comando show log** em intervalos regulares, cada 20 minutos a uma hora, a fim impedir a perda do log devendo proteger a reutilização.
- O que quer que acontece, inscreva o **comando show tech dos** membros afetados do grupo (GMs) e dos server chaves (KSs), e examine a saída do **comando show ip route em global** e cada roteamento virtual e transmissão (VRF) envolveram, se alguns são exigidos.

- Use a sincronização do Network Time Protocol (NTP) o pulso de disparo entre todos os dispositivos que são debugados. Permita timestamps do milissegundo (milissegundo) para ambos debugam e mensagens de registro:

```
GM1#debug crypto gdoi gm rekey ?
all-levels All levels
detail Detail level
error Error level
event Event level
packet Packet level
terse Terse level
```

- Certifique-se que os show command outputs (resultado do comando show) são timestamped.

```
Router#terminal exec prompt timestamp
```

- Quando você recolhe os show command outputs (resultado do comando show) para o controle aplanam eventos ou os contadores planos dos dados, recolhem sempre interações múltiplas da mesma saída.

## Pesquise defeitos o estabelecimento de IKE

Quando o processo de registro começa primeiramente, GMs e KSs negociam sessões do Internet Key Exchange (IKE) a fim proteger o tráfego GDOI.

- No GM, certifique-se do IKE esteja estabelecido com sucesso:

```
gm1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst src state conn-id status
172.16.1.9 172.16.1.1 GDOI_REKEY 1068 ACTIVE
172.16.1.1 172.16.1.9 GDOI_IDLE 1067 ACTIVE
```

**Note:** O estado GDOI\_IDLE, que é a base do registro, cronometra para fora rapidamente e desaparece, porque não é precisado anymore após o registro inicial.

- No KS, você deve ver:

```
ks1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst src state conn-id status
172.16.1.1 172.16.1.9 GDOI_IDLE 1001 ACTIVE
```

**Note:** A sessão do rekey aparece somente quando necessária no KS.

Termine estas etapas se você não alcança esse estado:

- Para a introspecção sobre a causa da falha, verifique a saída deste comando:

```
router# show crypto isakmp statistics
```

- Se a etapa precedente não é útil, você pode obter introspecções do nível de protocolo se você permite o IKE usual debuga:

```
router# debug crypto isakmp
```

Notas:

- \* Mesmo que o IKE seja usado, não é usado na porta UDP/500 usual, mas um pouco em UDP/848.
- \* Se você encontra uma edição neste nível, forneça debuga para KS e o GM afetado.
- Devido à dependência em sigs de Rivest-Shamir-Adleman (RSA) para o grupo rekeys, o KS **deve ter uma** chave RSA configurada, e deve ter o mesmo nome que esse especificado na configuração de grupo.

A fim verificar isto, incorpore este comando:

```
ks1# show crypto key mypubkey rsa
```

## Pesquise defeitos o registro inicial

No GM, a fim verificar o status de registro, examine a saída deste comando:

```
gm1# show crypto gdoi | i Registration status
Registration status : Registered
gm1#
```

Se a saída indica qualquer coisa a não ser **registrado**, incorpore estes comandos:

### No GMs:

- Shut down cripto-permitiu relações.  
**Caution:** Espera-se que o gerenciamento fora de banda está permitido.

- Permita estes debuga:

```
gm1# debug crypto gdoi infra packet
gm1# debug crypto gdoi gm packet
```

- Permita debuga no lado KS (veja a próxima seção).
- Quando o KS debuga esteja pronto, o unshut cripto-permitido conecta, e espera para o registro (a fim acelerar o processo, emita o comando **cripto claro do gdoi no GM**).

### No KSs:

- Verifique a presença da chave RSA no KS:

```
ks1# show crypto key mypubkey rsa
```

- Permita estes debugs:

```
ks1# debug crypto gdoi infra packet
ks1# debug crypto gdoi ks packet
```

## Pesquise defeitos edições Política-relacionadas

### O problema de política ocorre antes do registro (a política relativa do Falha-fim)

Esta edição afeta somente GMs, assim que recolha esta saída do GM:

```
gm1# show crypto ruleset
```

**Note:** No Cisco IOS XE<sup>?</sup>, esta saída está sempre vazia desde a classificação de pacote de informação no não feito no software.

A saída do **comando show tech** do dispositivo afetado fornece o resto da informação requerida.

### O problema de política ocorre registro do CARGO, e refere-se a política global que é empurrada

Há geralmente duas maneiras que este problema manifesta:

- O KS não pode empurrar as políticas para o GM.
- Há um aplicativo parcial da política entre o GMs.

A fim ajudar a pesquisar defeitos uma ou outra edição, termine estas etapas:

1. No GM afetado, recolha esta saída:

```
gm1# show crypto gdoi acl
gm1# show crypto ruleset
```

2. Permita estes debugs no GM:

```
gm1# debug crypto gdoi infra packet
gm1# debug crypto gdoi gm acls packet
```

3. No KS a que os registros afetados GM, recolhem esta saída:

```
ks1# show crypto gdoi ks members
ks1# show crypto gdoi ks policy
```

**Note:** A fim identificar a que KS o GM conecta, inscreva o comando `group crypto do gdoi` da mostra.

4. No mesmo KS, permita estes debugs:

```
ks1# debug crypto gdoi infra packet
ks1# debug crypto gdoi ks acis packet
```

5. Force o GM a registrar-se com este comando no GM:

```
clear crypto gdoi
```

O problema de política ocorre registro do CARGO, e refere-se a fusão da política global e o Local cancela

Esta edição manifesta-se geralmente sob a forma das mensagens que indicam que um pacote criptografado esteve recebido para que as políticas local indicam que não se supõe ser cifrado e vice-versa. Todos os dados pedidos na seção anterior e na saída do comando `show tech` são exigidos neste caso.

## Pesquise defeitos Rekey edições

No GMs:

- Recolha estes debugs:

```
gm1# debug crypto gdoi infra packet
gm1# debug crypto gdoi gm packet
gm1# debug crypto gdoi gm rekey packet
```

- Incorpore este comando a fim verificar que o GM ainda tem uma associação de segurança IKE (SA) do tipo GDOI\_REKEY:

```
gm1# show crypto isakmp sa
```

No KSs:

- Recolha o comando `show crypto key mypubkey rsa output` de CADA KS. As chaves são



esperadas ser **idênticas**.

- Entre nestes debuga a fim ver o que ocorre no KS:

```
ks1# debug crypto gdoi infra packet
ks1# debug crypto gdoi ks packet
ks1# debug crypto gdoi ks rekey packet
```

## Pesquise defeitos a Anti-repetição com base no período (TBAR)

A característica TBAR exige o tempo-mantimento através dos grupos, e exige consequentemente os pulsos de disparo do pseudo-tempo de GMs ser resynced constantemente. Isto é de disparo durante rekey ou cada duas horas, qualquer vem primeiramente.

**Note:** Toda a saída e debuga deve ser recolhida ao mesmo tempo de GMs e de KS de modo que possam ser correlacionados apropriadamente.

A fim investigar as edições que ocorrem neste nível, recolha esta saída.

- No GMs:

```
gm1# show crypto gdoi
gm1# show crypto gdoi replay
```

- No KS:

```
ks1# show crypto gdoi ks members
ks1# show crypto gdoi ks replay
```

A fim investigar TBAR quemantém-se em mais maneira dinâmica, permita estes debuga:

- No GM:

```
gm1# debug crypto gdoi gm rekey packet
gm1# debug crypto gdoi replay packet (verbosity might need to be lowered)
```

- No KS:

```
ks1# debug crypto gdoi ks rekey packet
ks1# debug crypto gdoi replay packet (verbosity might need to be lowered)
```

Até à data da Versão do IOS 15.2(3)T de Cisoc, a capacidade para gravar erros TBAR foi adicionada, que facilita manchar estes erros. No GM, use este comando a fim verificar se há algum erro TBAR:

```
R103-GM#show crypto gdoi gm replay
Anti-replay Information For Group GETVPN:
Timebased Replay:
```

```
Replay Value          : 512.11 secs
Input Packets          : 0           Output Packets          : 0
Input Error Packets    : 0           Output Error Packets    : 0
Time Sync Error        : 0           Max time delta          : 0.00secs
```

TBAR Error History (sampled at 10pak/min):

No TBAR errors detected

Para obter mais informações sobre de como pesquisar defeitos edições TBAR, refira a [falha baseada tempo da Anti-repetição](#).

## Pesquise defeitos a Redundância KS

A cooperativa (CAPOEIRA) estabelece uma sessão de IKE a fim proteger uma comunicação dos interKSs, assim que a técnica de Troubleshooting descrita previamente para o estabelecimento de IKE é aplicável aqui também.

o Troubleshooting Capoeira-específico compreende verificações da saída deste comando em todo o KSs envolveu:

```
ks# show crypto gdoi ks coop
```

**Note:** A maioria de erro comum feito com desenvolvimento da CAPOEIRA KSs é esquecer importar a mesma chave RSA (privado e público) para o grupo em todo o KSs. Isto causa problemas durante rekeys. A fim verificar e comparar chaves públicas entre KSs, compare a saída do comando **show crypto key mypubkey rsa** de cada KS.

Se o Troubleshooting do nível de protocolo é exigido, permita isto debugam em todo o KSs envolveu:

```
ks# debug crypto gdoi ks coop packet
```

## FAQ

### Por que você vê este % do ajuste do Mensagem de Erro “rekey a autenticação rejeitada”?

Você vê este Mensagem de Erro quando você configura o KS depois que esta linha está adicionada:

```
ks# debug crypto gdoi ks coop packet
```

A razão para este Mensagem de Erro é geralmente porque a chave etiquetada GETVPN\_KEYS não existe. A fim fixar isto, crie uma chave com a etiqueta correta usando o comando:

```
ks# debug crypto gdoi ks coop packet
```

**Note:** Adicionar a palavra-chave exportable na extremidade se este é um desenvolvimento da CAPOEIRA e importe então a mesma chave no outro KS

## **Pode um roteador configurado como KS para um grupo GETVPN igualmente para funcionar como um GM para o mesmos grupo?**

Não. Todas as disposições GETVPN exigem um KS dedicado que não possa participar como um GM para os mesmos grupos. Esta característica não é apoiada, porque adicionando a funcionalidade GM a KS com todas as interações possíveis como a criptografia, o roteamento, o QoS, etc., não é ótima para a saúde deste dispositivo de rede crucial. Deve estar disponível em todas as vezes para que o desenvolvimento inteiro GETVPN trabalhe.

## **Informações Relacionadas**

- [Transporte cifrado grupo VPN \(GET VPN\) - Cisco Systems](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)