

# GETVPN排除故障指南

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[GETVPN故障排除方法](#)

[参考拓扑](#)

[参考配置](#)

[术语](#)

[日志设备准备和其他最佳实践](#)

[排除故障GETVPN控制层面问题](#)

[控制层面调试最佳实践](#)

[GETVPN控制层面故障排除工具](#)

[GETVPN显示命令](#)

[GETVPN系统消息](#)

[全局crypto和GDOI调试](#)

[GDOI条件调试](#)

[GDOI事件跟踪](#)

[GETVPN控制层面Checkpoint和常见问题](#)

[小屋设置和策略创建](#)

[IKE设置](#)

[注册、策略下载和SA安装](#)

[重新生成密钥](#)

[控制层面中继检查](#)

[控制层面信息包分段问题](#)

[GDOI互操作性问题](#)

[排除故障GETVPN数据层面问题](#)

[GETVPN数据层面故障排除工具](#)

[加密/解密计数器](#)

[Netflow](#)

[DSCP/IP优先级标记](#)

[嵌入式数据包捕获](#)

[Cisco IOS XE数据包踪迹](#)

[GETVPN数据层面常见问题](#)

[通用的IPsec Dataplane问题](#)

[已知问题](#)

[排除故障在运行Cisco IOS XE的平台的GETVPN](#)

[故障排除命令](#)

[ASR1000常见问题](#)

[IPsec策略安装失败\(连续再登记\)](#)

[普通的迁移/升级问题](#)

[ASR1000 TBAR限制](#)

[ISR4x00分类问题](#)

[相关信息](#)

## 简介

本文打算提交结构故障排除方法和有用工具帮助识别和隔离分组加密传输VPN (GETVPN)问题和提供可能的解决方案。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- GETVPN
  - [正式GETVPN配置指南](#)
  - [正式GETVPN设计和实施指南](#)
- 系统日志服务器使用

### 使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

## GETVPN故障排除方法

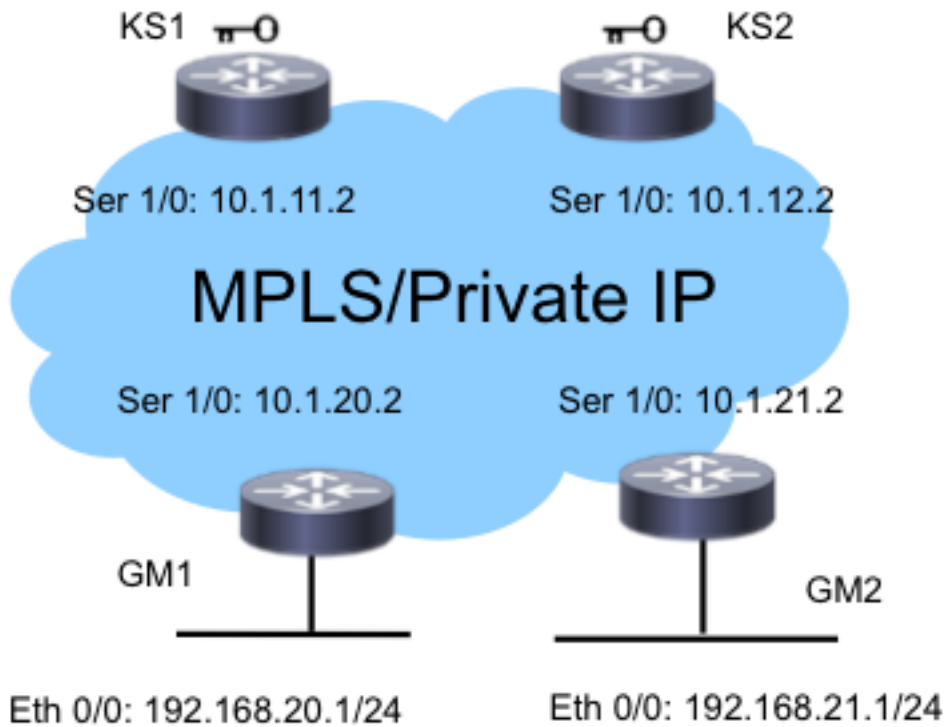
如同复杂技术问题多数故障排除，密钥是能离析问题一个特定功能、子系统或者组件。GETVPN解决方案包括一定数量的功能组件，特别地：

- Internet Key Exchange (IKE) -用于在组成员(GM)和关键服务器(KS)之间和在合作协议(小屋) KSs中为了验证和保护控制层面。
- 分组域解释(GDOI) -用于KS的协议为了分配组密钥，并且提供关键服务例如请重新生成密钥对所有GM。
- 小屋-用于KSs的协议为了彼此通信和提供冗余。
- 报头保存-在为端到端数据流交付保留原始信息包报头的隧道模式的IPsec。
- 时间根据反重放(TBAR) -用于组密钥环境的重播检测机制。

它也提供故障排除工具广泛的为了缓和排除故障进程。了解是重要的哪些工具是可用的，并且，当他们为每个故障排除任务时是适当的。当排除故障时，它总是一个好想法从最少插入的方法开始，以便生产环境没有负面影响。对此结构故障排除的密钥是能中止问题到控制或数据层面问题。如果跟随协议或数据流并且使用被提交此处为了检查点的多种工具他们，您能执行此。

### 参考拓扑

此GETVPN拓扑和编址方案使用在其余此故障排除文档中。



## 参考配置

### • KS1

```
crypto gdoi group G1
identity number 3333
server local
rekey authenmypubkeyrsa get
rekey transport unicast
sa ipsec 1
profile gdoi-p
match address ipv4ENCPOL
address ipv4 10.1.11.2
redundancy
local priority 10
peer address ipv4 10.1.12.2
```

### • GM1

```
crypto gdoi group G1
identity number 3333
server address ipv4 10.1.11.2
server address ipv4 10.1.12.2
!
crypto map gm_map 10 gdoi
set group G1
!
interface Serial1/0
crypto map gm_map
```

**Note:**KS2和GM2配置为简要起见没有包括此处。

## 术语

- **KS** -关键服务器
- **GM** -组成员
- **小屋**-合作协议
- **TBAR** -时间根据反重放
- **KEK** -密钥加密密钥
- **TEK** -数据流加密密钥

## 日志设备准备和其他最佳实践

在您开始排除故障前，请保证您准备日志设备如描述此处。也列出得一些最佳实践此处：

- 检查路由器空闲内存容量，并且配置logging buffered debugging对大价值若可能(10 MB或更多)。
- 禁用记录日志到控制台、监视器和系统日志服务器。
- 定期获取操作日志缓冲区内容用show log命令，每20分钟对1小时，为了防止由于日志的损耗缓冲重新使用。
- 什么发生，请输入show tech命令从受影响的GM和KSs，并且检查输出show ip route命令在全局，并且每虚拟路由和转发(VRF)介入，如果其中任一要求。
- 请使用网络时间协议(NTP)为了同步在调试的所有设备之间的时钟。启用调试和日志消息的毫秒(毫秒)时间戳：

```
service timestamps debug datetime msec
service timestamps log datetime msec
```

- 确保show命令输出是时间戳的。

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

- 当您收集控制层面事件或数据层面计数器的时show命令输出，总是请收集同一输出的多迭代。

## 排除故障GETVPN控制层面问题

控制层面含义导致对策略和安全关联(SA)创建在GM的所有协议事件，以便他们准备加密和解密数据层面流量。某些GETVPN控制层面的关键检查点是：



## 控制层面调试最佳实践

这些故障排除最佳实践不是GETVPN特定;他们适用于几乎所有控制层面调试。跟随这些最佳实践为了保证多数有效故障排除是关键：

- 关闭控制台记录并且请使用操作日志缓冲区或Syslog为了收集调试。
- 请使用NTP为了同步在调试的所有设备的路由器时钟。
- 启用调试和日志消息的毫秒时间戳：

```
service timestamp debug datetime msec
service timestamp log datetime msec
```

- 确保show命令输出是时间戳的，以便他们可以关联与debug输出：

```
terminal exec prompt timestamp
```

- 若可能请使用条件调试在缩放环境。

## GETVPN控制层面故障排除工具

### GETVPN显示命令

通常，这些是您应该为几乎所有GETVPN问题收集的命令输出。

#### KS

```
show crypto gdoi
show crypto gdoi ks coop
show crypto gdoi ks members
show crypto gdoi ks rekey
show crypto gdoi ks policy
```

#### GM

```
show crypto eli
show crypto isakmp sa
show crypto gdoi
show crypto gdoi gm
show crypto gdoi gm rekey
```

### GETVPN系统消息

GETVPN为重大的协议事件和错误情况提供系统消息广泛的。当您执行GETVPN故障排除时，Syslog应该总是查找的第一个地方。

#### 普通的KS系统消息

##### Syslog 消息

*COOP\_CONFIG\_MISMATCH*  
*COOP\_KS\_ELECTION*  
*COOP\_KS\_REACH*  
*COOP\_KS\_TRANS\_TO\_PRI*  
*COOP\_KS\_UNAUTH*  
*COOP\_KS\_UNREACH*  
*KS\_GM\_REVOKED*  
*KS\_SEND\_MCAST\_REKEY*  
*KS\_SEND\_UNICAST\_REKEY*  
*KS\_UNAUTHORIZED*  
*UNAUTHORIZED\_IPADDR*

##### 说明

在主键服务器和辅助密钥服务器之间的配置不匹配。

本地关键服务器在组中参与了选择进程。

在已配置的合作社密钥服务器之间的可接通性恢复。

**本地关键服务器过渡到从是的一个主用角色辅助服务器在组中。**

设法的一个已授权远程服务器联系本地关键服务器在组中，可能认为一个敌对事件。

**在已配置的合作社密钥服务器之间的可接通性丢失，可能认为一个敌对事件。**

在期间请重新生成密钥协议，一个未授权的成员设法参加组，可能认为一个敌对事件。

**发送组播请重新生成密钥。**

**发送单播请重新生成密钥。**

在GDOI注册协议期间，一个未授权的成员设法参加组，可能认为一个敌对事件。

因为请求的设备未授权参加组，注册请求丢弃了。

## 普通的GM系统消息

### Syslog 消息

GM\_CLEAR\_REGISTER

GM\_CM\_ATTACH

GM\_CM\_DETACH

GM\_RE\_REGISTER

GM\_RECV\_REKEY

GM\_REGS\_COMPL

GM\_REKEY\_TRANS\_2\_MULTI

GM\_REKEY\_TRANS\_2\_UNI

PSEUDO\_TIME\_LARGE

REPLAY\_FAILED

### 说明

清楚crypto gdoi命令由本地组成员执行了。

加密映射为本地组成员附加。

加密映射为本地组member.&被分开了

也许超时或已经清除了为一组创建的SA IPsec。需要再注册到关键服务器

rekey接收。

完整的注册。

组成员有从使用单播的已转换的重新生成密钥机制到使用组播机制。

组成员有从使用组播的已转换的重新生成密钥机制到使用单播机制。

组成员接收与是主要与其自己的pseudotime不同的值的一pseudotime。

组成员或密钥服务器失败一反重放检查。

**Note:**用红色突出显示的消息是在GETVPN环境看到的最普通或最重大的消息。

## 全局crypto和GDOI调试

GETVPN调试分开：

### 1. 首先由您排除故障的设备。

```
F340.06.15-2900-18#debug cry gdoi ?
all-features    All features in GDOI
condition       GDOI Conditional Debugging
gm              Group Member
ks              Key Server
```

### 2. 由您排除故障的问题类型的秒钟。

```
GM1#debug cry gdoi gm ?
all-features    All Group Member features
infrastructure  GM Infrastructure
registration    GM messages related to registration
rekey           GM messagese related to Re-Key
replay          Anti Replay
```

### 3. 由需要启用的级别的第三调试。在版本15.1(3)T和以上，所有GDOI功能调试标准化有这些调试级别。这设计为了帮助排除故障与足够的调试粒度的大规模GETVPN环境。当您调试GETVPN问题时，使用适当的调试级别是重要的。通常，请从最低的调试级别开始，那是错误级别，并且增加调试粒度，当需要。

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

## GDOI条件调试

在Cisco IOS版本15.1(3)T中和以后，GDOI条件调试被添加为了帮助排除故障在一个大规模环境的GETVPN。那么所有互联网安全协会和密钥管理协议(ISAKMP)和GDOI调试可能用根据组或对端IP地址的一个有条件的过滤器当前触发。对于多数GETVPN问题，因为GDOI调试只显示GDOI特定操作，启用ISAKMP和GDOI调试用适当的有条件的过滤器是好的。为了使用ISAKMP和GDOI有条件调试，请完成这两个简单步骤：

1. 设置有条件的过滤器。
2. 照常启用相关ISAKMP和GDOI。

例如：

```
KS1# debug crypto gdoi condition peer add ipv4 10.1.20.2
% GDOI Debug Condition added.
```

```
KS1#
KS1# show crypto gdoi debug-condition
GDOI Conditional Filters:
Peer Address 10.1.20.2
Unmatched NOT set
```

```
KS1#debug crypto gdoi ks registration all-levels
GDOI Key Server Registration Debug level: (Packet, Detail, Event, Terse, Error)
```

**Note:**使用两ISAKMP和GDOI有条件调试，为了捉住也许没有有条件的过滤器信息的调试消息，例如IP地址在调试路径，**不匹配**标志可以启用。然而，因为能导致很多调试信息，必须小心地使用这。

## GDOI事件跟踪

这在版本15.1(3)T被添加了。事件追踪提供轻量、不间断工作的跟踪重大的GDOI事件的和错误。也有跟踪与traceback的退出PATH启用为异常条件。事件跟踪比传统Syslog能提供更多GETVPN事件历史记录信息。

默认情况下GDOI事件跟踪启用并且可以从跟踪缓冲区获取用**show monitor均等trace**命令。

```
GM1#show monitor event-trace gdoi ?
all Show all the traces in current buffer
back Show trace from this far back in the past
clock Show trace from a specific clock time/date
coop GDOI COOP Event Traces
exit GDOI Exit Traces
from-boot Show trace from this many seconds after booting
infra GDOI INFRA Event Traces
latest Show latest trace events since last display
merged Show entries in all event traces sorted by time
registration GDOI Registration event Traces
rekey GDOI Rekey event Traces
```

```
GM1#show monitor event-trace gdoi rekey all
*Nov 6 15:55:16.117: GDOI_REKEY_EVENT: ACK_SENT: From 10.1.12.2 to 10.1.13.2
with seq no 1 for the group G1
*Nov 6 15:55:16.117: GDOI_REKEY_EVENT: REKEY_RCVD: From 10.1.12.2 to 10.1.13.2
with seq no 1 for the group G1
*Nov 6 16:11:01.125: GDOI_REKEY_EVENT: ACK_SENT: From 10.1.12.2 to 10.1.13.2
with seq no 1 for the group G1
*Nov 6 16:11:01.125: GDOI_REKEY_EVENT: REKEY_RCVD: From 10.1.12.2 to 10.1.13.2
with seq no 1 for the group G1
```

默认情况下退出路径跟踪提供关于退出路径的详细信息，那是例外和错误情况，当traceback选项启用。traceback可能然后用于为了解码导致了退出路径情况的确切的代码顺序。请使用**详细信息**选项为了从跟踪缓冲区获取traceback：

```

GM1#show monitor event-trace gdoi exit all detail
*Nov 6 15:15:25.611: NULL_VALUE_FOUND:Invalid GROUP Name
-Traceback= 0xCA51318z 0xCA1F4DBz 0xC9B2707z 0xCA1ED4Ez 0x97EB018z
0x97EA960z 0x97E8D62z 0x97F3706z 0x97F3361z 0xA02684Ez
*Nov 6 15:15:25.611: MAP_NOT_APPLIED_IN_ANY_INTERFACE:
-Traceback= 0xCA51318z 0xCA46718z 0xCA1EF79z 0x97EB018z 0x97EA960z
0x97E8D62z 0x97F3706z 0x97F3361z 0xA02684Ez 0xA01FD52z
*Nov 6 15:15:25.650: NULL_VALUE_FOUND:NULL Parameters passed idb or ipaddress
when idb ipaddress is changed
-Traceback= 0xCA51318z 0xCA22430z 0xA09A8DCz 0xA09D8F6z 0xA0F280Fz
0xBA1D1F4z 0xBA1CACCz 0xBA1C881z 0xBA1C5BBz 0xA0F494Az

```

默认跟踪缓冲区大小是512个条目，并且这也许不是足够，如果问题断断续续。为了增加此默认trace条目大小，事件trace配置参数可以更改类似显示此处：

```

GM1#show monitor event-trace gdoi rekey parameters
Trace has 512 entries
Stacktrace is disabled by default

```

```

GM1#
GM1#config t
Enter configuration commands, one per line. End with CNTL/Z.
GM1(config)#monitor event-trace gdoi rekey size ?
<1-1000000> Number of entries in trace

```

## GETVPN控制层面Checkpoint和常见问题

这是某些GETVPN的普通控制飞机问题。要重申，控制层面定义作为要求的所有GETVPN功能组件为了启用dataplane加密和解密在GM。在高层次，这要求成功的GM注册、安全策略和SA下载/安装，并且随后的KEK/TEK重新生成密钥。

### 小屋设置和策略创建

为了检查和验证KS顺利地创建安全策略和相关的KEK/TEK，输入：

```

KS1#show crypto gdoi ks policy
Key Server Policy:
For group G1 (handle: 2147483650) server 10.1.11.2 (handle: 2147483650):

For group G1 (handle: 2147483650) server 10.1.12.2 (handle: 2147483651):

# of teks : 1 Seq num : 10
KEK POLICY (transport type : Unicast)
spi : 0x18864836BA888BCD1126671EEAFEB4C7
management alg : disabled encrypt alg : 3DES
crypto iv length : 8 key size : 24
orig life(sec): 1200 remaining life(sec): 528
sig hash algorithm : enabled sig key length : 162
sig size : 128
sig key name : key1

TEK POLICY (encaps : ENCAPS_TUNNEL)
spi : 0x91E3985A
access-list : ENCPOL
transform : esp-null esp-sha-hmac
alg key size : 0 sig key size : 20
orig life(sec) : 900 remaining life(sec) : 796
tek life(sec) : 2203 elapsed time(sec) : 1407
override life (sec): 0 antireplay window size: 4

```



Replay Value 442843.29 secs

与KS策略设置的一常见问题是，当有不同的策略配置在主要的和附属KSs之间时。这能导致无法预测的KS行为，并且此错误将报告：

```
KS1#show crypto gdoi ks policy
```

```
Key Server Policy:
```

```
For group G1 (handle: 2147483650) server 10.1.11.2 (handle: 2147483650):
```

```
For group G1 (handle: 2147483650) server 10.1.12.2 (handle: 2147483651):
```

```
# of teks : 1 Seq num : 10
```

```
KEK POLICY (transport type : Unicast)
```

```
spi : 0x18864836BA888BCD1126671EEAFEB4C7
```

```
management alg : disabled encrypt alg : 3DES
```

```
crypto iv length : 8 key size : 24
```

```
orig life(sec): 1200 remaining life(sec): 528
```

```
sig hash algorithm : enabled sig key length : 162
```

```
sig size : 128
```

```
sig key name : key1
```

```
TEK POLICY (encaps : ENCAPS_TUNNEL)
```

```
spi : 0x91E3985A
```

```
access-list : ENCPOL
```

```
transform : esp-null esp-sha-hmac
```

```
alg key size : 0 sig key size : 20
```

```
orig life(sec) : 900 remaining life(sec) : 796
```

```
tek life(sec) : 2203 elapsed time(sec) : 1407
```

```
override life (sec): 0 antireplay window size: 4
```

Replay Value 442843.29 secs

目前没有主要的和附属KSs之间的自动配置同步，因此必须手工纠正这些。

由于小屋是GETVPN的一关键(和几乎总是必须)配置，是关键正确地确保小屋工作，并且小屋KS角色正确：

```
KS1#show crypto gdoi ks coop
```

```
Crypto Gdoi Group Name :G1
```

```
Group handle: 2147483650, Local Key Server handle: 2147483650
```

```
Local Address: 10.1.11.2
```

```
Local Priority: 200
```

```
Local KS Role: Primary , Local KS Status: Alive
```

```
Local KS version: 1.0.4
```

```
Primary Timers:
```

```
Primary Refresh Policy Time: 20
```

```
Remaining Time: 10
```

```
Antireplay Sequence Number: 40
```

```
Peer Sessions:
```

```
Session 1:
```

```
Server handle: 2147483651
```

```
Peer Address: 10.1.12.2
```

```
Peer Version: 1.0.4
```

```
Peer Priority: 100
```

```
Peer KS Role: Secondary , Peer KS Status: Alive
```

```
Antireplay Sequence Number: 0
```

```
IKE status: Established
```

Counters:

```
Ann msgs sent: 31
Ann msgs sent with reply request: 2
Ann msgs rcv: 64
Ann msgs rcv with reply request: 1
Packet sent drops: 7
Packet Recv drops: 0
Total bytes sent: 20887
Total bytes rcv: 40244
```

在一个功能小屋设置，应该观察此协议流：

## IKE Exchange > ANN以小屋优先级交换>小屋从主要的选择> ANN到第二KS (策略、GM数据库和密钥)

当小屋不正确地时运转，或者，如果有小屋已分解，例如变为的多个KSs必须为排除故障收集主要的KS，这些调试：

```
debug crypto isakmp
debug crypto gdoi ks coop all-levels
show crypto isakmp sa
show crypto gdoi ks coop
```

## IKE设置

成功的IKE交换为GETVPN要求为了巩固随后的策略和SA下载的控制通道。在成功的IKE交换结束时，此IKE SA创建：

```
GM1#show crypto isa sa
IPv4 Crypto ISAKMP SA
dst src state conn-id status
10.1.13.2 10.1.11.2 GDOI_REKEY          1075 ACTIVE
10.1.11.2 10.1.13.2 GDOI_IDLE           1074 ACTIVE

IPv6 Crypto ISAKMP SA

GM1#
```

**Note:**一旦最初的IKE交换完成，随后的策略和密钥从KS将推送到与使用的GM GDOI\_REKEY SA。因此，当他们超时，没有为GDOI\_REKEY SA重新生成密钥;当他们的寿命超时，他们消失。然而，应该总是有在GM的SA GDOI\_REKEY为了它能接收重新生成密钥。

GETVPN的IKE交换是没有与用于传统点到点IPSec隧道的IKE不同，因此故障排除方法依然是同样。必须收集这些调试为了排除故障IKE验证问题：

```
debug crypto isakmp
debug crypto isakmp error
debug crypto isakmp detail (hidden command, if detailed isakmp exchange information
is needed)
debug crypto isakmp packet (hidden command, if packet level isakmp information is needed)
```

## 注册、策略下载和SA安装

一旦IKE验证成功，GM向KS登记。当这正确地发生时，这些系统消息预计被看到：

```
%GDOI-5-GM_REKEY_TRANS_2_UNI: Group G1 transitioned to Unicast Rekey.  
%GDOI-5-SA_KEK_UPDATED: SA KEK was updated  
%GDOI-5-SA_TEK_UPDATED: SA TEK was updated  
%GDOI-5-GM_REGS_COMPL: Registration to KS 10.1.12.2 complete for group G1 using  
address 10.1.13.2  
%GDOI-5-GM_INSTALL_POLICIES_SUCCESS: SUCCESS: Installation of Reg/Rekey policies  
from KS 10.1.12.2 for group G1 & gm identity 10.1.13.2
```

策略和密钥可以用此命令验证：

```
GM1#show crypto gdoi
```

```
GROUP INFORMATION
```

```
Group Name : G1  
Group Identity : 3333  
Crypto Path : ipv4  
Key Management Path : ipv4  
Rekeys received : 1  
IPSec SA Direction : Both
```

```
Group Server list : 10.1.11.2  
10.1.12.2
```

```
Group member : 10.1.13.2 vrf: None
```

```
Version : 1.0.4
```

```
Registration status : Registered
```

```
Registered with : 10.1.12.2
```

```
Re-registers in : 139 sec
```

```
Succeeded registration: 1
```

```
Attempted registration: 1
```

```
Last rekey from : 10.1.11.2
```

```
Last rekey seq num : 0
```

```
Unicast rekey received: 1
```

```
Rekey ACKs sent : 1
```

```
Rekey Rcvd(hh:mm:ss) : 00:05:20
```

```
allowable rekey cipher: any
```

```
allowable rekey hash : any
```

```
allowable transformtag: any ESP
```

```
Rekeys cumulative
```

```
Total received : 1
```

```
After latest register : 1
```

```
Rekey Acks sents : 1
```

```
ACL Downloaded From KS 10.1.11.2:
```

```
access-list deny icmp any any
```

```
access-list deny eigrp any any
```

```
access-list deny ip any 224.0.0.0 0.255.255.255
```

```
access-list deny ip 224.0.0.0 0.255.255.255 any
```

```
access-list deny udp any port = 848 any port = 848
```

```
access-list permit ip any any
```

```
KEK POLICY:
```

```
Rekey Transport Type : Unicast
```

```
Lifetime (secs) : 878
```

```
Encrypt Algorithm : 3DES
```

```
Key Size : 192
```

```
Sig Hash Algorithm : HMAC_AUTH_SHA
```

```
Sig Key Length (bits) : 1024
```

```
TEK POLICY for the current KS-Policy ACEs Downloaded:
```

```
Serial1/0:
```

IPsec SA:  
spi: 0x8BF147EF(2347845615)  
transform: esp-3des esp-sha-hmac  
sa timing:remaining key lifetime (sec): (200)  
Anti-Replay(Time Based) : 4 sec interval

GM1#  
GM1#  
GM1#show crypto ipsec sa

interface: Serial1/0  
Crypto map tag: gmlmap, local addr 10.1.13.2

protected vrf: (none)  
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)  
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)  
current\_peer 0.0.0.0 port 848  
PERMIT, flags={}  
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0  
#pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0  
#pkts compressed: 0, #pkts decompressed: 0  
#pkts not compressed: 0, #pkts compr. failed: 0  
#pkts not decompressed: 0, #pkts decompress failed: 0  
#send errors 0, #recv errors 0

local crypto endpt.: 10.1.13.2, remote crypto endpt.: 0.0.0.0  
path mtu 1500, ip mtu 1500, ip mtu idb Serial1/0  
current outbound spi: 0x0(0)  
PFS (Y/N): N, DH group: none

local crypto endpt.: 10.1.13.2, remote crypto endpt.: 0.0.0.0  
path mtu 1500, ip mtu 1500, ip mtu idb Serial1/0  
current outbound spi: 0x8BF147EF(2347845615)  
PFS (Y/N): N, DH group: none

inbound esp sas:  
**spi: 0x8BF147EF(2347845615)**  
transform: esp-3des esp-sha-hmac ,  
in use settings ={Tunnel, }  
conn id: 1, flow\_id: SW:1, sibling\_flags 80000040, crypto map: gmlmap  
sa timing: remaining key lifetime (sec): (192)  
Kilobyte Volume Rekey has been disabled  
IV size: 8 bytes  
replay detection support: Y replay window size: 4  
Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcg sas:

outbound esp sas:  
**spi: 0x8BF147EF(2347845615)**  
transform: esp-3des esp-sha-hmac ,  
in use settings ={Tunnel, }  
conn id: 2, flow\_id: SW:2, sibling\_flags 80000040, crypto map: gmlmap  
sa timing: remaining key lifetime (sec): (192)  
Kilobyte Volume Rekey has been disabled  
IV size: 8 bytes  
replay detection support: Y replay window size: 4  
Status: ACTIVE(ACTIVE)

outbound ah sas:

```
outbound pcp sas:
GM1#
```

**Note:**使用GETVPN，入站和出站SAS使用同样SPI。

使用GETVPN注册和策略安装问题类型，这些调试是需要的为了排除故障：

```
debug crypto isakmp (KS and GM)
debug crypto gdoi ks registration all-levels (KS)
debug crypto gdoi gm registration all-level (GM)
debug crypto engine (GM only)
show crypto eli detail (multiple iterations on GM)
```

**Note:**另外的调试可能根据这些输出结果要求。

因为GETVPN注册在GM重新加载之后典型地发生，此EEM脚本也许是有用为了收集这些调试：

```
debug crypto isakmp (KS and GM)
debug crypto gdoi ks registration all-levels (KS)
debug crypto gdoi gm registration all-level (GM)
debug crypto engine (GM only)
show crypto eli detail (multiple iterations on GM)
```

## 重新生成密钥

一旦GM注册对KS，并且GETVPN网络适当地被建立，主要的KS对发送负责重新生成密钥消息对所有GM注册对它。重新生成密钥消息用于为了同步所有策略、密钥和pseudotimes在GM。重新生成密钥信息可以通过单播或组播方法传送。

当重新生成密钥信息传送时，此系统消息在KS被看到：

```
debug crypto isakmp (KS and GM)
debug crypto gdoi ks registration all-levels (KS)
debug crypto gdoi gm registration all-level (GM)
debug crypto engine (GM only)
show crypto eli detail (multiple iterations on GM)
```

在GM，这是被看到的Syslog，当接收重新生成密钥时：

```
debug crypto isakmp (KS and GM)
debug crypto gdoi ks registration all-levels (KS)
debug crypto gdoi gm registration all-level (GM)
debug crypto engine (GM only)
show crypto eli detail (multiple iterations on GM)
```

## RSA密钥对需求为在KS重新生成密钥

重新生成密钥功能要求RSA密钥出现在KS的。KS提供RSA密钥对的公共密钥给GM通过此安全信道在注册时。KS然后签署GDOI发送的消息对与私有RSA密钥的GM在GDOI SIG有效负载。GM收到GDOI消息并且使用公共RSA密钥为了验证消息。在KS和GM之间的消息加密与KEK，也被分配对GM在注册时。一旦注册完成，随后重新生成密钥加密与KEK并且签字与私有RSA密钥。

如果RSA密钥是在KS的没有存在GM注册时，此消息出现在Syslog：

```
debug crypto isakmp (KS and GM)
debug crypto gdoi ks registration all-levels (KS)
debug crypto gdoi gm registration all-level (GM)
debug crypto engine (GM only)
show crypto eli detail (multiple iterations on GM)
```

当密钥不是存在KS时，GM第一次注册，但是重新生成密钥从KS失效。最终在GM的现有密钥超时，并且再再注册。

```
debug crypto isakmp (KS and GM)
debug crypto gdoi ks registration all-levels (KS)
debug crypto gdoi gm registration all-level (GM)
debug crypto engine (GM only)
show crypto eli detail (multiple iterations on GM)
```

因为RSA密钥对用于为了签署重新生成密钥消息，他们必须是相同的在主要的和所有附属KSs之间。这保证在一主要的KS失败期间，重新生成密钥发送由第二KS (新的主要的KS)能由GM适当地仍然验证。当它生成在主要的KS时的RSA密钥对，必须创建密钥对与可导出选项，以便他们可以导出到所有第二KSs为了符合此要求。

### 重新生成密钥故障排除

KEK/TEK重新生成密钥失败是在用户部署遇到的其中一最普通的GETVPN问题。排除故障重新生成密钥问题应该遵从重新生成密钥步骤如概述此处：

#### 1. 重新生成密钥获得发送由KS ?

这可以由%GDOI-5-KS\_SEND\_UNICAST\_REKEY系统消息的observion更加准确地检查或用此命令：

```
KS1#show crypto gdoi ks rekey
Group G1 (Unicast)
Number of Rekeys sent           : 341
Number of Rekeys retransmitted  : 0
KEK rekey lifetime (sec) : 1200
Remaining lifetime (sec) : 894
Retransmit period : 10
Number of retransmissions : 5
IPSec SA 1 lifetime (sec) : 900
Remaining lifetime (sec) : 405
```

编号重新生成密钥重新传输是预示的重新生成密钥KS没接收的确认数据包并且可能请重新生成密钥问题。记住GDOI重新生成密钥用途UDP作为不可靠的传输传输机制，因此一些重新生成密钥丢包也许根据基础传输网络的可靠性预计，但是趋势增加重新生成密钥重新传输应该总是调查。

更多详细的每GM重新生成密钥统计信息可能也获取。这典型地是寻找可能性的第一个地方重新生成密钥问题。

```
KS1#show crypto gdoi ks members

Group Member Information :
```

```
Number of rekeys sent for group G1 : 346

Group Member ID : 10.1.14.2 GM Version: 1.0.4
Group ID : 3333
Group Name : G1
Key Server ID : 10.1.11.2
  Rekeys sent      : 346
Rekeys retries : 0
Rekey Acks Rcvd : 346
Rekey Acks missed : 0

Sent seq num : 2 1 2 1
Rcvd seq num : 2 1 2 1
```

```
Group Member ID : 10.1.13.2 GM Version: 1.0.4
Group ID : 3333
Group Name : G1
Key Server ID : 10.1.12.2
  Rekeys sent      : 340
Rekeys retries : 0
Rekey Acks Rcvd : 340
Rekey Acks missed : 0

Sent seq num : 2 1 2 1
Rcvd seq num : 2 1 2 1
```

## 2. 在基础结构网络重新生成密钥数据包被传送？

在转接网络应该跟随沿重新生成密钥转发路径的标准的IP故障排除为了保证重新生成密钥数据包没有丢弃在KS和GM之间。使用的一些普通的故障排除工具这里是输入/输出访问控制列表 (ACL)、Netflow和数据包捕获转接网络的。

## 3. 重新生成密钥GDOI进程为重新生成密钥处理的数据包伸手可及的距离？

检查GM重新生成密钥统计信息：

```
G1#show crypto gdoi gm rekey
Group G1 (Unicast)
Number of Rekeys received (cumulative) : 340
Number of Rekeys received after registration : 340
Number of Rekey Acks sent : 340
```

## 4. 重新生成密钥确认数据包返回对KS？

遵从步骤1至3为了跟踪从GM的重新生成密钥确认数据包回到KS。

### Muticast重新生成密钥

组播在这些方面重新生成密钥是与单播不同重新生成密钥：

- 因为组播用于为了传输这些重新生成密钥从KS的数据包到GM，KS不需要复制重新生成密钥数据包。KS只发送重新生成密钥数据包的一复制，并且他们在支持组播的网络复制。
- 没有组播的确认机制重新生成密钥，因此，如果GM不是收到重新生成密钥数据包，KS不了解它，并且从其GM数据库不会删除GM。并且，因为没有确认，KS永远将重传根据其的重新生成密钥数据包重新生成密钥重新传输配置。

当重新生成密钥在GM时，没有接收最编解码器的组播重新生成密钥问题是。能有此的一定数量的

可能的原因，例如：

- 在组播路由结构内的数据包传输问题
- 端到端组播路由没有在网络内启用

排除故障与组播的一个问题的第一步重新生成密钥将发现是否请重新生成密钥工作，当交换从组播到单播方法。

一旦识别问题是特定组播请重新生成密钥，验证KS发送重新生成密钥对指定的组播地址。

```
GM1#show crypto gdoi gm rekey
Group G1 (Unicast)
Number of Rekeys received (cumulative) : 340
Number of Rekeys received after registration : 340
Number of Rekey Acks sent : 340
```

测试KS和GM之间的组播连接与互联网控制消息协议(ICMP)请求对组播地址。是组播组的一部分的所有GM应该应答到ping。保证ICMP从此测验的KS加密策略被排除。

```
KS1#ping 226.1.1.1
```

```
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 226.1.1.1, timeout is 2 seconds:
```

```
Reply to request 0 from 10.1.21.2, 44 ms
```

如果组播ping测试失败，则必须执行组播故障排除，是在范围本文外面。

## 控制层面中继检查

### 症状

当客户升级他们的GM对新的Cisco IOS版本时，他们也许体验KEK重新生成密钥有在Syslog观察的此消息的失败：

```
KS1#ping 226.1.1.1
```

```
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 226.1.1.1, timeout is 2 seconds:
```

```
Reply to request 0 from 10.1.21.2, 44 ms
```

此行为是由互操作性问题引起的介绍与为控制层面消息被添加的反重放检查。特别地，运行更旧的代码的KS将重置KEK重新生成密钥序号到1，并且这将由运行新的代码的GM丢弃，当作为被重赛的请重新生成密钥数据包的解释。欲了解更详细的信息，请参阅Cisco Bug ID [CSCta05809](#) (GETVPN：GETVPN控制面板易于重赛的)和[GETVPN配置限制](#)。

### 背景

使用GETVPN，控制层面消息能传播时间敏感的信息为了提供基于时间的反重放检查服务。所以，这些消息要求反重放保护为了保证时间accuracy。这些消息是：

- 重新生成密钥消息从KS到GM
- 关在KSs之间的通知消息



作为此反重放保护实施一部分，当TBAR启用时，序号检查被添加为了保护被重赛的消息，以及pseudotime检查。

## 解决方案

为了解决此问题，必须升级GM和KS到Cisco IOS版本，在控制层面重播检查功能后。使用新型Cisco IOS代码，KS不重新设置序号到1 KEK的重新生成密钥，反而继续使用当前序号和只重置TEK的序号重新生成密钥。

这些Cisco IOS版本有重播检查功能：

- 12.4(15)T10
- 12.4(22)T3
- 12.4(24)T2
- 15.0(1)M及以后

## 其他重赛相关问题

- 小屋失败由于失败重播的ANN消息检查(Cisco Bug ID [CSCtc52655](#))

## 调试控制层面重播失败

对于其他控制层面重播失败，请收集此信息并且确保时代同步在KS和GM之间。

- 从GM和KS的Syslog
- ISAKMP调试
- GDOI调试(请重新生成密钥和重播)从KS和GM

## 控制层面信息包分段问题

使用GETVPN，控制层面信息包分段是常见问题，并且能在这两个方案之一中表明自己，当控制层面数据包足够时大他们将要求IP分段：

- GETVPN小屋通告数据包
- GETVPN重新生成密钥数据包

## 小屋通告数据包

小屋通告数据包传播GM数据库信息和能因而增长大在一大GETVPN部署。从经验，包括1500+ GM将生产通告的数据包的GETVPN网络大于18024个字节，是Cisco IOS默认巨大缓冲区大小。当这发生时，KS不能分配足够大的缓冲区传送有此错误的ANN数据包：

```
KS1#ping 226.1.1.1
```

```
Type escape sequence to abort.
```

```
Sending 1, 100-byte ICMP Echos to 226.1.1.1, timeout is 2 seconds:
```

```
Reply to request 0 from 10.1.21.2, 44 ms
```

为了纠正此情况，推荐此缓冲调节：

```
KS1#ping 226.1.1.1
```

```
Type escape sequence to abort.
```

```
Sending 1, 100-byte ICMP Echos to 226.1.1.1, timeout is 2 seconds:
```

```
Reply to request 0 from 10.1.21.2, 44 ms
```

## 重新生成密钥数据包

GETVPN重新生成密钥数据包可以也超出典型1500 IP最大转换单元(MTU)大小，当加密策略大时，例如包括8+线路访问控制条目的策略(ACE)在加密ACL。

## 分段问题和识别

在两个上一个方案，GETVPN一定能适当地传送，并且收到被分段的UDP数据包为了小屋或GDOI请重新生成密钥适当地工作。IP分段可以是在一些网络环境的一问题。例如，包括等价多路径的网络(ECMP)转发架构和在转发架构的一些设备要求被分段的IP信息包的虚拟重组，例如虚拟分段重组(VFR)。

为了识别问题，请检查在我们怀疑的设备的重组错误被分段的UDP 848数据包没有适当地接收：

```
KS1#show ip traffic | section Frags
```

```
Frag: 10 reassembled, 3 timeouts, 0 couldn't reassemble
```

```
0 fragmented, 0 fragments, 0 couldn't fragment
```

如果重组超时继续增加，请使用**error命令的debug ip**为了确认丢弃是否是rekey/COOP数据包流的一部分。一旦确认，应该执行正常IP转发排除故障为了隔离在也许已经丢弃了数据包的转发架构的确切的设备。一些常用的工具包括：

- 数据包捕获
- 流量转发统计信息
- 安全功能统计信息(防火墙，IPS)
- VFR统计信息

## GDOI互操作性问题

多种互操作性问题多年来找到与GETVPN，并且注意Cisco IOS版本版本在KS和GM之间和在互操作性问题的KSs中是关键。

其他著名的GETVPN互操作性问题是：

- 控制层面中继检查
- [GETVPN KEK重新生成密钥行为崔凡吉莱](#)
- Cisco Bug ID [CSCub42920](#) - GETVPN : KS不能验证哈希重新生成密钥从上一个GM版本的ACK
- Cisco Bug ID [CSCuw48400](#) (无法的GetVPN GM注册或重新生成密钥发生故障- SIG HASH >默认SHA-1)

## GETVPN IOS升级步骤

此Cisco IOS升级做法，当Cisco IOS代码升级在GETVPN环境时，需要被执行应该仿效：

1. 首先请升级第二KS并且等待，直到小屋KS选择完成。
2. 重复所有附属KSs的Step1。
3. 升级主要的KS。
4. 升级GM。

## 排除故障GETVPN数据层面问题

与控制层面问题比较，GETVPN数据层面问题是由于某种原因GM有进行策略和的密钥dataplane加密和解密的问题，但是端到端数据流运输流量不运作。大多GETVPN的dataplane问题与转发通用的IPsec关连，并且不是GETVPN特定。因此描述的大多数故障排除方法此处适用于通用的IPsec dataplane问题。

使用加密问题(基于组的或成对地通道)，排除故障问题和离析问题数据路径的特定的部分是重要的。特别地，描述的故障排除方法此处打算帮助您应答这些问题：

- 哪个设备是罪犯-加密路由器或解密路由器？
- 在哪个方向发生的问题-入口或出口？

## GETVPN数据层面故障排除工具

IPsec dataplane故障排除是非常与那不同为控制层面。使用dataplane，通常没有您能运行的调试，或者安全至少运行在生产环境。因此故障排除在可帮助跟踪沿转发路径的数据包的不同的计数器 and 流量统计取决于。想法是能开发一套检查点来帮助隔离数据包也许丢弃如显示此处的地方：



这是一些数据层面调试工具：

- 访问列表
- IP优先级核算
- Netflow
- 接口计数器
- crypto计数器
- IP Cisco Express Forwarding (CEF)全局和每功能丢弃计数器
- 嵌入式数据包捕获(EPC)
- 数据层面调试(IP数据包和CEF调试)

数据路径的检查点在前一个镜像可以验证与这些工具：

### 加密GM

- 入口LAN接口
  - 输入ACL
  - 入口Netflow
  - [嵌入式数据包捕获](#)

\_\_\_\_\_输入优先次序记帐

- 加密引擎

- show crypto ipsec sa
  - show crypto ipsec sa详细信息
  - show crypto engine加速器统计信息

- 出口广域网接口

- 输出NetFlow

- [嵌入式数据包捕获](#)

\_\_\_\_\_输出优先次序记帐

## 解密GM

- 入口广域网接口

- 输入ACL

- 入口Netflow

- [嵌入式数据包捕获](#)

\_\_\_\_\_输入优先次序记帐

- 加密引擎

- show crypto ipsec sa
  - show crypto ipsec sa详细信息
  - show crypto engine加速器统计信息

- 出口LAN接口

- 输出NetFlow

- 嵌入式数据包捕获

返回路径跟随同一通信流。以下部分有这些dataplane工具一些示例在使用中。

## 加密/解密计数器

在路由器的加密/解密计数器根据IPsec流。不幸地这不工作良好与GETVPN，因为GETVPN典型地实施加密一切的“permit ip any any”加密策略。因此，如果问题为一些流而不是所有只发生，这些计数器可以是有些难使用为了正确地估计，如果数据包加密或解密，当有工作的足够重大的后台流量时。

```
GM1#show crypto ipsec sa | in encrypt|decrypt
#pkts encaps: 100, #pkts encrypt: 100, #pkts digest: 100
#pkts decaps: 100, #pkts decrypt: 100, #pkts verify: 100
```

## Netflow

Netflow可以用于为了监控入口和出口流量在两GM。与GETVPN permit ip any any策略的注意，encrypted流量将是聚合，并且不提供单个流的信息。单个流的信息然后将需要收集与DSCP/precedence标记描述的以后。

在本例中，-100计数ping的Netflow从在GM1后的一台主机对在GM2后的一台主机在多种检查点显示。

## 加密GM

Netflow配置：

```
interface Ethernet0/0
description LAN
ip address 192.168.13.1 255.255.255.0
ip flow ingress
ip pim sparse-dense-mode
!
interface Serial1/0
description WAN interface
ip address 10.1.13.2 255.255.255.252
ip flow egress
ip pim sparse-dense-mode
crypto map gmlmap
```

NetFlow输出：

```
GM1#show ip cache flow | be SrcIf
SrcIf SrcIPAddress DstIf DstIPAddress Pr SrcP DstP Pkts
Et0/0 192.168.13.2 Se1/0* 192.168.14.2 32 8DE1 6523 100
Et0/0 192.168.13.2 Se1/0 192.168.14.2 01 0000 0800 100
GM1#
```

**Note:**在上一个输出中，\*表示出口流量。第一行显示出口加密流量(与协议0x32 = ESP)在广域网接口外面和点击LAN接口的第二个线路入口ICMP流量。

## 解密GM

配置：

```
interface Ethernet0/0
description LAN interface
ip address 192.168.14.1 255.255.255.0
ip flow egress
ip pim sparse-dense-mode
!
interface Serial1/0
description WAN interface
ip address 10.1.14.2 255.255.255.252
ip flow ingress
ip pim sparse-dense-mode
crypto map gmlmap
```

NetFlow输出：

```
GM2#show ip cache flow | be SrcIf
SrcIf SrcIPAddress DstIf DstIPAddress Pr SrcP DstP Pkts
Se1/0 192.168.13.2 Et0/0 192.168.14.2 32 8DE1 6523 100
Se1/0 192.168.13.2 Et0/0* 192.168.14.2 01 0000 0800 100
GM2#
```

## DSCP/IP优先级标记

与排除故障加密问题的挑战是，一旦数据包加密您丢失可见性到有效负载，是什么加密应该执行，并且那使困难跟踪一特定的Ip流的数据包。有两种方式寻址此限制当谈到排除故障IPsec问题：

- 请使用ESP-NULL作为IPsec转换。IPsec仍然执行ESP封装，但是不加密应用对有效负载，因此他们是可视在数据包捕获。
- 标记—Ip流用根据他们的L3/L4特性的一个唯一差分服务代码点/precedence标记。

ESP-NULL要求在两个通道端点的更改和经常没有允许根据客户安全策略。所以，思科典型地推荐指示使用的DSCP/precedence。

#### DSCP/Precedence参考图

Tos (十六进制)	ToS(Decimal)	IP 优先级	DSCP	二进制
0xE0	224	7网络控制	56个CS7	111 00000
0xC0	192	6互连网络控制	48个CS6	110 00000
0xB8	184	5关键	46个EF	101 11000
0xA0	160		40个CS5	101 00000
0x88	136	4闪存覆盖	34个AF41	100 01000
0x80	128		32个CS4	100 00000
0x68	104	3闪存	26个AF31	011 01000
0x60	96		24个CS3	011 00000
0x48	72	2立即	18个AF21	010 01000
0x40	64		16个CS2	010 00000
0x20	32	1优先级	8个CS1	001 00000
0x00	0	0惯例	0 Dflt	000 00000

#### 用DSCP/Precedence标记数据包

这些方法典型地用于为了用特定DSCP/Precedence标记标记数据包。

#### PBR

```
interface Ethernet1/0
ip policy route-map mark
!
access-list 150 permit ip host 172.16.1.2 host 172.16.254.2
!
route-map mark permit 10
match ip address 150
set ip precedence flash-override
```

#### MQC

```
class-map match-all my_flow
match access-group 150
!
policy-map marking
class my_flow
set ip precedence 4
!
interface Ethernet1/0
service-policy input marking
```

#### 路由器Ping

```
G01-host#ping ip
Target IP address: 192.168.14.2
```

```
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
Type of service [0]: 136
...
<snip>
```

**Note:**它总是一个好想法监控正常通信流和DSCP/precedence配置文件，在您应用标记前，以便明显通信流是唯一。

## 箴言报标记的信息包

## IP优先级核算

```
interface Ethernet0/0
ip address 192.168.1.2 255.255.255.0
ip accounting precedence input
```

```
middle_router#show interface precedence
Ethernet0/0
Input
Precedence 4: 100 packets, 17400 bytes
```

## 接口ACL

```
middle_router#show access-list 144
Extended IP access list 144
10 permit ip any any precedence routine
20 permit ip any any precedence priority
30 permit ip any any precedence immediate
40 permit ip any any precedence flash
50 permit ip any any precedence flash-override (100 matches)
60 permit ip any any precedence critical
70 permit ip any any precedence internet (1 match)
80 permit ip any any precedence network
```

## [嵌入式数据包捕获](#)

嵌入式数据包捕获(EPC)是获取数据包的有用工具在接口级为了识别，如果数据包到达了一个特定设备。切记良好EPC工作对于明文流量，但是它可以是挑战，当获取数据包加密。所以技术类似以前讨论的DSCP/precedence标记或其他IP字符，例如IP数据包的长度，必须与EPC一起用于为了做排除故障更加有效。

## Cisco IOS XE数据包踪迹

这是跟踪运行Cisco IOS XE，例如CSR1000v，ASR1000的所有平台和ISR4451-X的功能转发路径的有用的功能。

## GETVPN数据层面常见问题

排除故障GETVPN的IPsec dataplane是主要没有与排除故障传统点到点IPsec dataplane问题不同，有两例外由于GETVPN这些唯一dataplane属性。

## 时间根据反重放失败

在GETVPN网络中，因为成对地不再有通道，TBAR失败可以经常是难排除故障。为了排除故障GETVPN TBAR失败，请完成这些步骤：

1. 识别哪数据包丢弃的归结于TBAR失败和随后识别加密的GM。

在版本15.3(2)T之前，TBAR失败Syslog没有打印失败的数据包的源地址，因此这使非常困难识别哪数据包失败。这在版本15.3(2)T和以上显著改善，Cisco IOS打印此：

```
middle_router#show access-list 144
Extended IP access list 144
10 permit ip any any precedence routine
20 permit ip any any precedence priority
30 permit ip any any precedence immediate
40 permit ip any any precedence flash
50 permit ip any any precedence flash-override (100 matches)
60 permit ip any any precedence critical
70 permit ip any any precedence internet (1 match)
80 permit ip any any precedence network
```

TBAR历史记录在此版本也实现：

```
GM2#show crypto gdoi gm replay
Anti-replay Information For Group G1:
Timebased Replay:
Replay Value : 621388.66 secs
Input Packets : 0 Output Packets : 0
Input Error Packets : 2 Output Error Packets : 0
Time Sync Error : 0 Max time delta : 0.00 secs
```

```
TBAR Error History (sampled at 10pak/min):
19:29:32.081 EST Wed Nov 13 2013: src=192.168.13.2; my_pst=620051.84 secs;
peer_pst=619767.09 secs; win=4
```

**Note:**以前被提及的增强从那以后实现在Cisco IOS XE由Cisco Bug ID [CSCun49335](#)和在Cisco IOS由Cisco Bug ID [CSCub91811](#)。

对于没有此功能的Cisco IOS版本，`debug crypto gdoi gm`重播详细信息能也提供此信息，虽然此调试打印所有流量的(不仅被丢弃的数据包TBAR信息由于TBAR失败)，因此在生产环境也许不可行运行。

```
GM2#show crypto gdoi gm replay
Anti-replay Information For Group G1:
Timebased Replay:
Replay Value : 621388.66 secs
Input Packets : 0 Output Packets : 0
Input Error Packets : 2 Output Error Packets : 0
Time Sync Error : 0 Max time delta : 0.00 secs
```

```
TBAR Error History (sampled at 10pak/min):
19:29:32.081 EST Wed Nov 13 2013: src=192.168.13.2; my_pst=620051.84 secs;
```



```
peer_pst=619767.09 secs; win=4
```

- 一旦数据包的来源识别，您应该能查找加密的GM。然后，应该为所有潜在的pseudotime偏差监控在加密的和解密的GM的pseudotimestamp。要执行此的最佳方法是同步GM和KS对NTP和周期地收集pseudotime信息用在所有的一个参考系统时钟为了确定问题是否由在GM的时滞引起。

## GM1

```
GM1#show crypto gdoi gm replay
```

```
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
```

```
Time source is hardware calendar, *21:06:26.469 EST Wed Nov 13 2013
```

```
Anti-replay Information For Group G1:
```

```
Timebased Replay:
```

```
Replay Value : 625866.26 secs
```

```
Input Packets : 0 Output Packets : 0
```

```
Input Error Packets : 0 Output Error Packets : 0
```

```
Time Sync Error : 0 Max time delta : 0.00 secs
```

## GM2

```
GM2#show crypto gdoi gm replay
```

```
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
```

```
Time source is hardware calendar, *21:06:26.743 EST Wed Nov 13 2013
```

```
Anti-replay Information For Group G1:
```

```
Timebased Replay:
```

```
Replay Value : 625866.51 secs
```

```
Input Packets : 4 Output Packets : 4
```

```
Input Error Packets : 2 Output Error Packets : 0
```

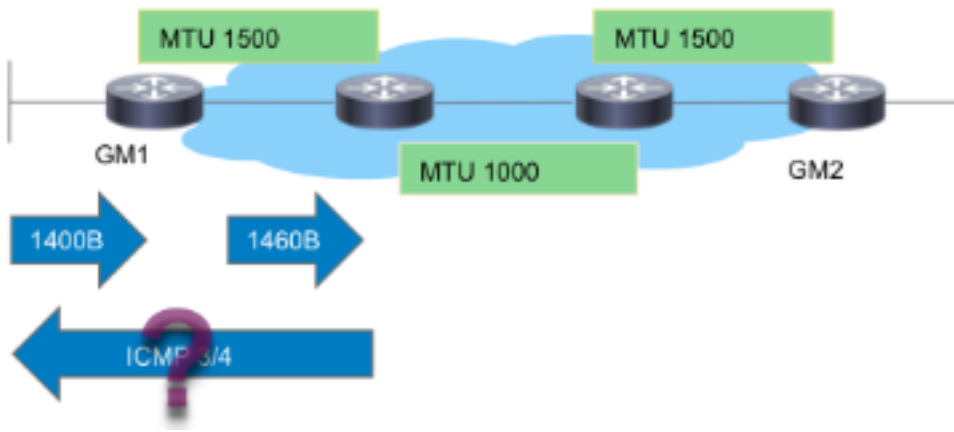
```
Time Sync Error : 0 Max time delta : 0.00 secs
```

在前一个示例中，如果pseudotime (如表示的是由重播值)是较大不同在GM之间，当输出捕获与同一参考时间时，然后问题可以归因于时滞。

**Note:**在Cisco聚集的服务路由器1000系列平台上，由于平台体系结构，在Quantum流处理器(QFP)的数据路径实际上是指计数的pseudotime瞬间壁钟。当墙壁时钟时间更改由于NTP同步时，这创建与TBAR的问题。此问题描述与Cisco Bug ID [CSCum37911](#)。

## PMTUD和GETVPN报头保存

使用GETVPN，路径MTU发现(PMTUD)不工作在加密的和解密的GM之间和有不要分段(DF)位集的大数据包能获得黑洞。原因这不工作归结于GETVPN数据源/目的地址在封装报头的ESP保留的报头保存。这在此镜像表示：



当镜像显示，PMTUD划分与与此流的GETVPN：

1. 大型数据包在加密的GM1到达。
2. POST加密ESP数据包转发在GM1外面并且传送往目的地。
3. 如果有与1400个字节IP MTU的转接链路，ESP数据包将丢弃，并且ICMP 3/4数据包太大信息将传送往信息包来源，是数据包的来源。
4. ICMP3/4数据包是丢弃的由于从GETVPN加密策略在结束时没排除的ICMP或者丢弃的主机，因为不知道什么ESP数据包(未经鉴定的有效负载)。

总之，PMTUD不与GETVPN今天一起使用。为了在此问题附近工作，思科推荐这些步骤：

1. 实现“ip tcp adjust-mss”为了减少TCP数据包分段大小罐子顺序o适应加密开销和最低路径MTU在转接网络。
2. 清除数据包的DF位，他们在加密的GM到达为了避免PMTUD。

## 通用的IPsec Dataplane问题

大多数IPsec dataplane故障排除是类似排除故障传统点到点IPSec隧道。其中一个常见问题是%CRYPTO-4-RECVD\_PKT\_MAC\_ERR。请参阅[与Ping损耗的Syslog "%CRYPTO-4-RECVD\\_PKT\\_MAC\\_ERR:"](#)错误消息在IPSec隧道故障排除关于更多故障排除详细信息。

## 已知问题

此消息可以生成，当不匹配在SADB的一个SPI的IPSec信息包接收时。请参阅Cisco Bug ID [CSCtd47420](#) - GETVPN -为pkt报告的CRYPTO-4-RECVD\_PKT\_NOT\_IPSEC不匹配流。例如：

```
GM2#show crypto gdoi gm replay
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is hardware calendar, *21:06:26.743 EST Wed Nov 13 2013
```

```
Anti-replay Information For Group G1:
Timebased Replay:
Replay Value           : 625866.51 secs
Input Packets : 4 Output Packets : 4
Input Error Packets : 2 Output Error Packets : 0
Time Sync Error : 0 Max time delta : 0.00 secs
```

此消息应该是%CRYPTO-4-RECVD\_PKT\_INV\_SPI，是什么得到报告为传统IPsec以及在某个硬件平台例如ASR。此表面问题由Cisco Bug ID [CSCup80547](#)修复：在报告CRYPTO-4-RECVD\_PKT\_NOT\_IPSEC的错误ESP的朴。

**Note:** 这些消息能有时看上去由于另一GETVPN bug [CSCup34371](#) : 在TEK重新生成密钥后, GETVPN GM停止decrypting流量。

在这种情况下, GM不能解密GETVPN流量, 虽然有一有效IPsec SA在SADB (重新生成密钥的SA)。当SA从SADB, 超时和删除问题消失。因为TEK重新生成密钥事先, 执行此问题导致重大的中断。例如, 中断可以是22分钟一旦TEK寿命7200秒。请参阅bug说明关于应该符合为了遇到此bug的确切的情况。

## 排除故障在运行Cisco IOS XE的平台的GETVPN

### 故障排除命令

运行Cisco IOS XE的平台有平台特定的实施和经常要求GETVPN问题的平台特定的调试。这是典型地用于的命令列表为了排除故障在这些平台的GETVPN :

**show crypto eli全部**

**显示平台软件ipsec策略统计信息**

**显示平台软件ipsec fp活动库存**

**显示平台硬件qfp活动功能ipsec spd全部**

**显示平台硬件qfp活动统计信息丢弃结算**

**显示平台硬件qfp活动功能ipsec数据丢弃结算**

**show crypto ipsec sa**

**show crypto gdoi**

**显示内部crypto的ipsec**

**debug crypto ipsec**

**debug crypto ipsec错误**

**debug crypto ipsec状态**

**debug crypto ipsec消息**

**debug crypto ipsec HW req**

**在下debug crypto gdoi gm详细信息**

**debug crypto gdoi gm重新生成密钥详细信息**

### ASR1000常见问题

**IPsec策略安装失败(连续再登记)**

如果加密引擎不支持接收的IPsec策略或算法ASR1000 GM也许继续注册到关键服务器。例如，在Nitrox根据ASR平台(例如ASR1002)，套件B或SHA2不支持策略，并且这能导致连续再登记症状。

## 普通的迁移/升级问题

### ASR1000 TBAR限制

在ASR1000平台上，Cisco Bug ID [CSCum37911](#)修正介绍在不支持少于20秒的TBAR时间的此平台的一个限制。请参阅[限制关于在IOS-XE的GETVPN。](#)

打开此增强bug放松此限制，Cisco Bug ID [CSCuq25476](#) - ASR1k需要支持GETVPN TBAR窗口大小少于20秒。

**更新：**此限制从那以后放松了以Cisco Bug ID的[CSCur57558](#)修正，并且它不再是一个限制用XE3.10.5、XE3.13.2及以后代码。

并且为在Cisco IOS XE平台运行的GM请注释，(ASR1k或ISR4k)，我们极力推荐设备运行版本以此问题的修正，如果TBAR启用;Cisco Bug ID [CSCut91647](#) -在IOS-XE的GETVPN：GM不正确地丢弃数据包由于TBAR失败。

### ISR4x00分类问题

返回在拒绝策略忽略的ISR4x00平台被找到。关于详细信息，请参阅Cisco Bug ID [CSCut14355](#) - GETVPN - ISR4300 GM忽略拒绝策略。

## 相关信息

- [分组加密传输VPN \(GET VPN\) - Cisco系统](#)
- [技术支持和文档 - Cisco Systems](#)