系统日志“%CRYPTO-4-RECVD_PKT_MAC_ERR：”错误消息Ping Loss Over IPsec隧道故障排除

目录

简介

本文档介绍如何解决IPsec隧道上与系统日志中的“%CRYPTO-4-RECVD_PKT_MAC_ERR”消息结合的ping丢失问题，如方框所示：

```
May 23 11:41:38.139 GMT: %CRYPTO-4-RECVD_PKT_MAC_ERR:
decrypt: mac verify failed for connection
id=2989 local=172.16.200.18 remote=172.16.204.18 spi=999CD43B
seqno=00071328
```

这种下降的一小部分被视为正常。但是，由于此问题导致的高丢包率可能会影响服务，并且可能需要网络运营商的注意。请注意，系统日志中报告的这些消息的速率限制为30秒，因此，单个日志消息并不总是表示只丢弃了一个数据包。要获得这些丢包的准确计数，请发出命令**show crypto ipsec sa detail**，并查看日志中显示的连接ID旁的SA。在SA计数器中，**pkts verify failed** 错误计数器会**计由于消息验证代码(MAC)验证失败而导致的总丢包数。**

```
interface: GigabitEthernet0/1
Crypto map tag: MPLSWanGREVPN, local addr 172.16.204.18

protected vrf: (none)
local ident (addr/mask/prot/port): (172.16.204.18/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.16.205.18/255.255.255.255/47/0)
current_peer 172.16.205.18 port 500
PERMIT, flags={origin_is_acl,}
#pkts encaps: 51810, #pkts encrypt: 51810, #pkts digest: 51810
#pkts decaps: 44468, #pkts decrypt: 44468, #pkts verify: 44468
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
```

```
#pkts not decompressed: 0, #pkts decompress failed: 0
#pkts no sa (send) 0, #pkts invalid sa (rcv) 0
#pkts encaps failed (send) 0, #pkts decaps failed (rcv) 0
#pkts invalid prot (recv) 0, #pkts verify failed: 8
#pkts invalid identity (recv) 0, #pkts invalid len (rcv) 0
#pkts replay rollover (send): 0, #pkts replay rollover (rcv) 0
#pkts replay failed (rcv): 0
#pkts internal err (send): 0, #pkts internal err (recv) 0

local crypto endpt.: 172.16.204.18, remote crypto endpt.: 172.16.205.18
path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet0/1
current outbound spi: 0xD660992C(3596654892)

inbound esp sas:
spi: 0x999CD43B(2577191995)
transform: esp-3des esp-sha-hmac ,
in use settings ={Transport, }
conn id: 2989, flow_id: AIM-VPN/SSL-3:2989, sibling_flags 80000006,
crypto map: MPLSWanGREVPN
sa timing: remaining key lifetime (k/sec): (4257518/24564)
IV size: 8 bytes
replay detection support: N
Status: ACTIVE

outbound esp sas:
spi: 0xD660992C(3596654892)
transform: esp-3des esp-sha-hmac ,
in use settings ={Transport, }
conn id: 2990, flow_id: AIM-VPN/SSL-3:2990, sibling_flags 80000006,
crypto map: MPLSWanGREVPN
sa timing: remaining key lifetime (k/sec): (4199729/24564)
IV size: 8 bytes
replay detection support: N
Status: ACTIVE
```

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档中的信息基于对Cisco IOS® 15.1(4)M4版进行的测试。虽然尚未测试，但脚本和配置应与早期的Cisco IOS软件版本配合使用，因为两个小程序都使用EEM 3.0版(IOS 12.4(22)T或更高版本支持)。

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。

功能信息

“%CRYPTO-4-RECVD_PKT_MAC_ERR:decrypt:"表示收到的加密数据包未通过MAC验证。此验证

是配置的身份验证转换集的结果：

```
Router (config)# crypto ipsec transform transform-1 esp-aes 256 esp-md5-hmac
```

在上例中，*"esp-aes 256"*将加密算法定义为256位AES，*"esp-md5"*将MD5（HMAC变体）定义为用于身份验证的哈希算法。MD5等散列算法通常用于提供文件内容的数字指纹。数字指纹通常用于确保文件未被入侵者或病毒更改。因此，此错误消息的出现通常意味着：

- 使用错误的密钥加密或解密数据包。此错误非常罕见，可能是由软件Bug引起的。

— 或 —

- 数据包在传输过程中被篡改。此错误可能是由于电路不正常或恶意事件。

故障排除方法

由于此错误消息通常是由数据包损坏引起的，因此进行根本原因分析的唯一方法是使用EPC从两个隧道端点的WAN端获取完整的数据包捕获并进行比较。在获取捕获之前，最好确定触发这些日志的流量类型。在某些情况下，它可能是特定类型的流量；在其他情况下，它可能是随机的，但很容易被复制（例如每100个ping操作5-7次丢弃）。 在这种情况下，问题变得容易识别。识别触发器的最佳方法是使用DSCP标记标记测试流量并捕获数据包。DSCP值将复制到ESP报头，然后可以使用Wireshark进行过滤。此配置假设测试100个ping，可用于标记ICMP数据包：

```
ip access-list extended VPN_TRAFFIC
 permit icmp <source> <destination>
class-map match-all MARK
 match access-group name VPN_TRAFFIC
policy-map MARKING
 class MARK
   set dscp af21
```

此策略现在必须应用于加密路由器上收到清除流量的入口接口：

```
interface GigabitEthernet0/0
service-policy MARKING in
```

或者，您可能希望使用路由器生成的流量运行此测试。为此，您无法使用服务质量(QoS)标记数据包，但可以使用基于策略的路由(PBR)。

注意：要查找关键(5)DSCP标记，请使用Wireshark过滤器**ip.dsfield.dscp == 0x28**。

```
ip access-list extended VPN_TRAFFIC
 permit icmp <source> <destination>
route-map markicmp permit 10
match ip address vpn
set ip precedence critical
ip local policy route-map markicmp
```

为ICMP流量配置QoS标记后，您可以配置嵌入式数据包捕获：

```
Router(config)# ip access-list ext vpn_capo
Router(config)# permit ip host
```

```
Router(config)# permit ip host
```

```
Router(config)#  exit //the capture is only configured in enable mode.
Router# monitor capture buffer vpncap size 256 max-size 100 circular
Router# monitor capture buffer vpncap filter access-list vpn_capo
Router# monitor capture point ip cef capo fastEthernet 0/1 both
Router# monitor capture point associate capo vpncap
Router# monitor capture point start capo //starts the capture.
To stop replace the "start" keyword with "stop"
```

注意：此功能在思科IOS版本12.4(20)T中引入。有关EPC的详细信息，请参阅嵌入式数据包捕获。

使用数据包捕获来排除此类故障需要捕获整个数据包，而不仅仅是其一部分。在15.0(1)M之前的Cisco IOS版本中，EPC功能的缓冲区限制为512K，最大数据包大小限制为1024字节。为避免此限制，请升级到15.0(1)M或更高版本的代码，该代码现在支持捕获缓冲区大小为100M，最大数据包大小为9500字节。

如果每100次ping都能可靠地重现问题，则最坏的情况是安排维护窗口，以便仅允许ping流量作为受控测试并捕获。此过程只需几分钟，但在此期间确实会中断生产流量。如果使用QoS标记，则无需仅对ping数据包进行限制。要捕获一个缓冲区中的所有ping数据包，必须确保在高峰时段不执行测试。

如果问题不容易重现，您可以使用EEM脚本自动捕获数据包。理论是，将两端的捕获都启动到循环缓冲区中，然后使用EEM停止一端的捕获。EEM在停止捕获的同时，让它向对等体发送snmp陷阱，从而停止捕获。此过程可能会起作用。但是，如果负载过重，则第二台路由器可能反应不够快，无法停止捕获。优选受控测试。以下是将实施该流程的EEM脚本：

```
Receiver
========
event manager applet detect_bad_packet
event syslog pattern "RECVD_PKT_MAC_ERR"
action 1.0 cli command "enable"
action 2.0 cli command "monitor capture point stop test"
action 3.0 syslog msg "Packet corruption detected and capture stopped!"
action 4.0 snmp-trap intdata1 123456 strdata ""
```

```
Sender
======
event manager applet detect_bad_packet
event snmp-notification oid 1.3.6.1.4.1.9.10.91.1.2.3.1.9.
oid-val "123456" op eq src-ip-address 20.1.1.1
action 1.0 cli command "enable"
action 2.0 cli command "monitor capture point stop test"
action 3.0 syslog msg "Packet corruption detected and capture stopped!"
```

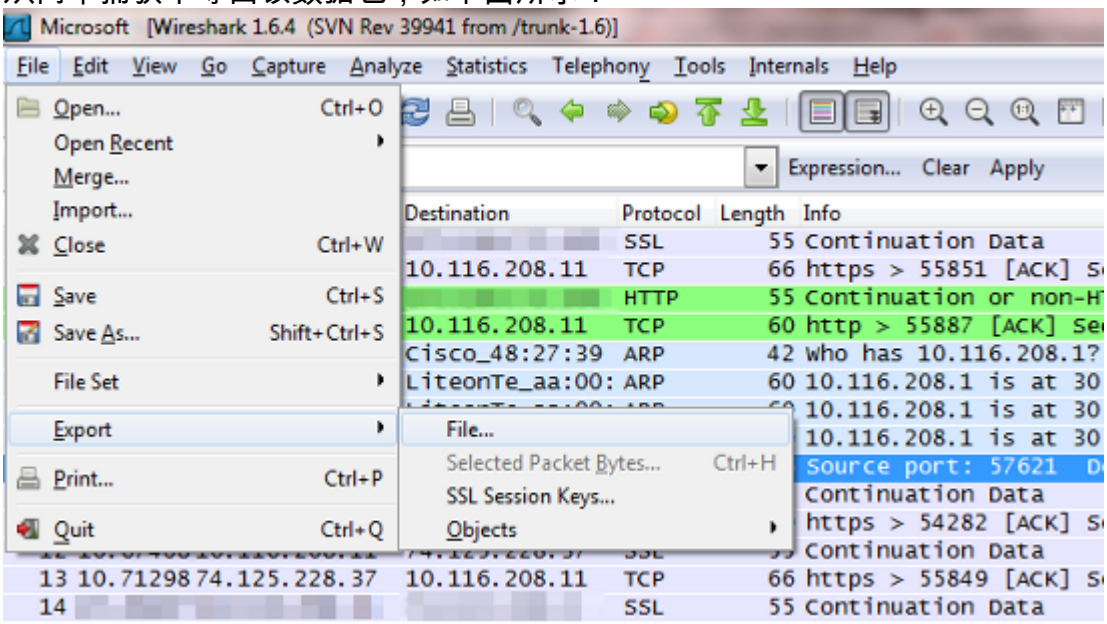
请注意，上一个框中的代码是经15.0(1)M测试的配置。在客户环境中实施之前，您可能希望使用客户使用的特定Cisco IOS版本对其进行测试。

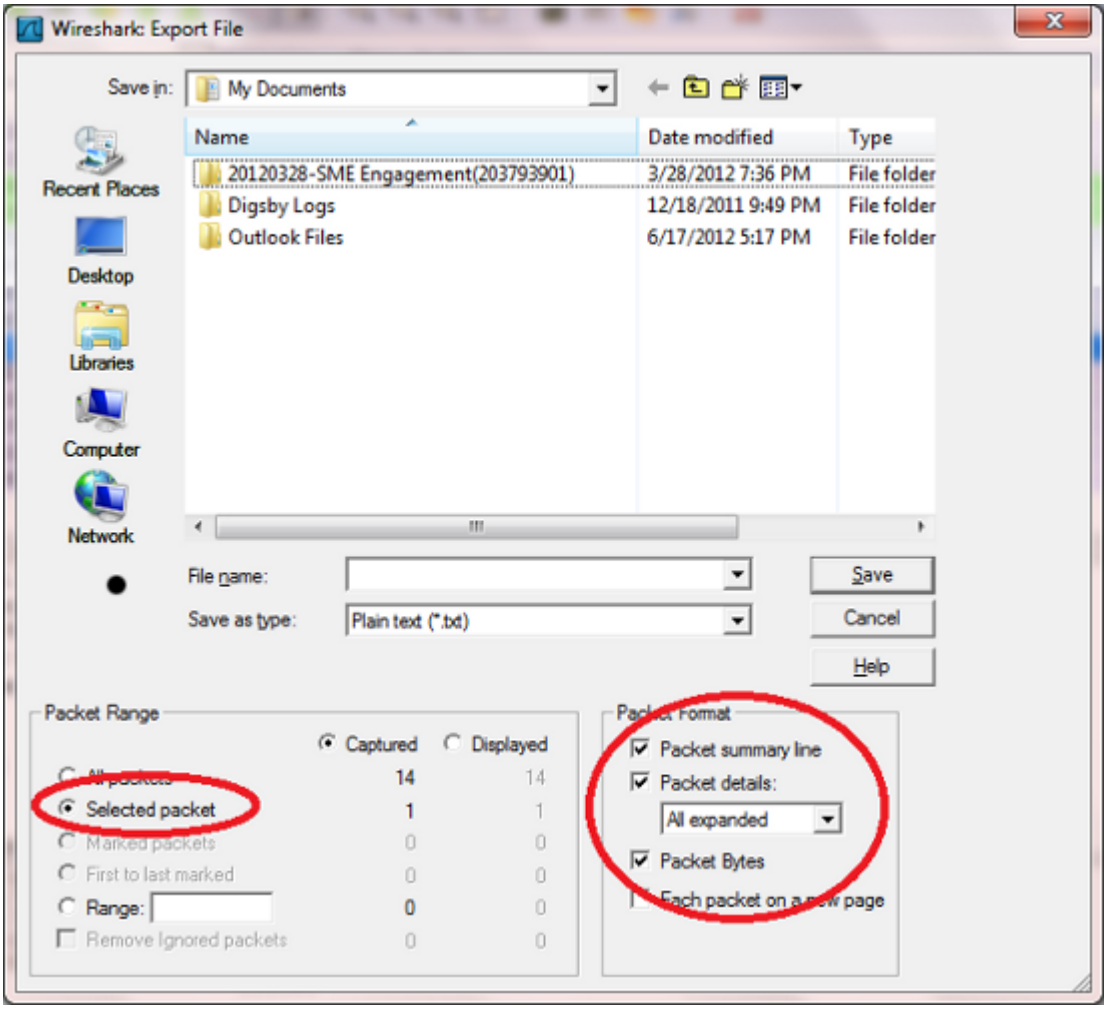
数据分析

1. 捕获完成后，使用TFTP将其导出到PC。
2. 使用网络协议分析器（如Wireshark）打开捕获。
3. 如果使用了QoS标记，请过滤出相应的数据包。

   ```
   ip.dsfield.dscp==0x08
   ```

   “0x08”特定于DSCP值AF21。如果使用不同的DSCP值，则可以从数据包捕获本身或从DSCP值转换图表列表获取正确的值。有关详细信息，请参阅DSCP和优先级值。
4. 识别从发送方捕获的ping丢弃，并在接收方和发送方捕获的数据包中查找该数据包。
5. 从两个捕获中导出该数据包，如下图所示：





6. 对两者进行二进制比较。如果两者相同，则传输中没有错误，Cisco IOS在接收端抛出错误的负数或在发送端使用错误的密钥。无论哪种情况，问题都是Cisco IOS Bug。如果数据包不同，则数据包在传输中被篡改。

下面是数据包在FC上离开加密引擎时的数据包：

```
*Mar  1 00:01:38.923: After encryption:
05F032D0: 45000088 00000000 E.......
05F032E0: FF3266F7 0A01201A 0A012031 7814619F .2fw.. ... 1x.a.
05F032F0: 00000001 DE9B4CEF ECD9178C 3E7A7F24 ....^.LolY..>z.$
05F03300: 83DCF16E 7FD64265 79F624FB 74D5AEF2 .\qn.VBeyv${tU.r
05F03310: 5EC0AC16 B1F9F3AB 89524205 A20C4E58 ^@,.1ys+.RB.".NX
05F03320: 09CE001B 70CC56AB 746D6A3A 63C2652B .N..pLV+tmj:cBe+
05F03330: 1992E8AF 2CE2A279 46367BDB 660854ED ..h/,b"yF6{[f.Tm
05F03340: 77B69453 83E47778 1470021F 09436285 w6.S.dwx.p...Cb.
05F03350: CB94AEF5 20A65B1F 480D86F6 125BA12E K..u &[.H..v.[!.
```

以下是在对等体上收到的相同数据包：

```
4F402C90:                   45000088 00000000          E.......
4F402CA0: FF3266F7 0A01201A 0A012031 7814619F .2fw.. ... 1x.a.
4F402CB0: 00000001 DE9B4CEF ECD9178C 3E7A7F24 ....^.LolY..>z.$
4F402CC0: 83DCF16E 7FD64265 79F624FB 74D5AEF2 .\qn.VBeyv${tU.r
4F402CD0: 5EC0AC16 B1F9F3AB 89524205 A20C4E58 ^@,.1ys+.RB.".NX
4F402CE0: 09CE001B 70CC56AB 00000000 00000000 .N..pLV+........
4F402CF0: 00000000 00000000 00000000 00000000 ................
4F402D00: 00000000 00000000 00000000 00000000 ................
4F402D10: 00000000 00000000 00000000 00000000 ................
```

此时，很可能是ISP问题，该组应参与故障排除。

常见问题

- Cisco Bug ID CSCed87408描述了83x上加密引擎的硬件问题，其中随机传出数据包在加密期间损坏，这会导致身份验证错误（在使用身份验证的情况下）和接收端丢包。请务必注意，您不会在83x上看到这些错误，而是在接收设备上看到。

- 有时运行旧代码的路由器会显示此错误。您可以升级到更新的代码版本(如15.1(4)M4)以解决此问题。

- 要验证问题是硬件还是软件问题，请禁用硬件加密。如果日志消息继续，则是软件问题。否则，RMA应解决问题。
  请记住，如果禁用硬件加密，会导致重负载VPN隧道的网络严重降级。因此，思科建议您在维护窗口期间尝试本文档中描述的过程。

相关信息

- 技术支持和文档 - Cisco Systems