Syslog "%CRYPTO−4−RECVD_PKT_MAC_ERR:" Error Message with Ping Loss Over IPsec Tunnel Troubleshooting

TAC **Document ID: 116085**

Contributed by Cisco TAC Engineers.
Oct 24, 2013

Contents

Introduction

This document describes how to resolve ping loss over an IPsec tunnel coupled with "%CRYPTO−4−RECVD_PKT_MAC_ERR" messages in the syslog as shown in the box:

```
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
```

A small percentage of such drops is considered normal. However, a high drop rate because of this problem can impact service and might require the network operator's attention. Note that these messages reported in the syslogs are rate limited at 30 second intervals, so a single log message does not always indicate that only a single packet got dropped. In order to obtain an accurate count of these drops, issue the command ***show crypto ipsec sa detail***, and look at the SA next to the connection ID seen in the logs. Among the SA counters, the ***pkts verify failed*** error counter accounts for the total packet drop due to the message authentication code (MAC) verification failure.

```
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
```

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on tests done with Cisco IOS® Release 15.1(4)M4. Although not yet tested, the scripts and configuration should work with earlier Cisco IOS software versions as well since both applets use EEM version 3.0 (which is supported in IOS version 12.4(22)T or above).

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.

Feature Information

The "%CRYPTO−4−RECVD_PKT_MAC_ERR: decrypt:" implies that an encrypted packet was received that failed the MAC verification. This verification is a result of the authentication transform set configured:

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

In the above example, *"esp−aes 256"* defines the encryption algorithm as 256−bit AES, and *"esp−md5"* defines the MD5 (HMAC variant) as the hash algorithm used for authentication. Hash algorithms like MD5 are typically used to provide a digital fingerprint of a file's contents. The digitial fingerprint is often used to ensure that the file has not been altered by an intruder or virus. Thus the occurence of this error message usually implies either:

- The wrong key was used to encrypt or decrypt the packet. This error is very rare and could be caused by a software bug.

  −OR−
- The packet was tampered with during transit. This error could be due to a dirty circuit or a hostile event.

Troubleshooting Methodology

Since this error message is typically caused by packet corruption, the only way to do a root−cause analysis is to use EPC in order to obtain complete packet captures from the WAN side on both tunnel end points and to compare them. Before you obtain the captures, it is best to identify what kind of traffic triggers these logs. In some cases, it can be a specific kind of traffic; in other cases, it might be random but easily reproduced (such as 5−7 drops every 100 pings). In such situations, the issue becomes slightly easier to identify. The best way to identify the trigger is to mark the test traffic with DSCP markings and to capture the packets. The DSCP value is copied to the ESP header and can then be filtered with Wireshark. This configuration, which assumes a test with 100 pings, can be used to mark the ICMP packets:

```
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
```

This policy must now be applied to the ingress interface where the clear traffic is received on the encrypting router:

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

Alternatively, you might want to run this test with router−generated traffic. For this, you are not able to use Quality of Service (QoS) to mark the packets, but you can use Policy−Based Routing (PBR).

Note: In order to locate critical (5) DSCP markings, use the Wireshark filter ***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
```

Once QoS marking is configured for your ICMP traffic, you can configure the embedded packet capture:

```
Router(config)# ip access−list ext vpn_capo
Router(config)# permit ip host <local> <peer>
Router(config)# permit ip host <peer> <local>
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"
```

Note: this feature was introduced in Cisco IOS Release 12.4(20)T. Refer to Embedded Packet Capture for more information regarding EPCs.

The use of a packet capture to troubleshoot this type of problem requires that the entire packet be captured, not just a portion of it. The EPC feature in Cisco IOS releases prior to 15.0(1)M has a buffer limit of 512K and a max packet size limit of 1024 bytes. In order to avoid this limitation, upgrade to 15.0(1)M or newer code, which now supports a capture buffer size of 100M with a max packet size of 9500 bytes.

If the issue can be reliably reproduced with every 100 count ping, the worst−case scenario is to schedule a maintenance window in order to allow only the ping traffic as a controlled test and take the captures. This process should take only a few minutes, but it does disrupt production traffic for that time. If you use QoS marking, you can eliminate the requirement to restrict packets only to pings. In order to capture all the ping packets in one buffer, you must ensure that the test is not conducted during peak hours.

If the issue is not easily reproduced, you can use an EEM script to automate the packet capture. The theory is that you start the captures on both sides into a circular buffer and use EEM to stop the capture on one side. At the same time the EEM stops the capture, have it send an snmp trap to the peer, which stops its capture. This process might work. But if the load is heavy, the second router might not react quickly enough to stop its capture. A controlled test is preferred. Here are the EEM scripts that will implement the process:

```
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!"
```

Note that the code in the previous box is a configuration tested with 15.0(1)M. You might want to test it with the specific Cisco IOS version your customer uses before you implement it in the customer environment.

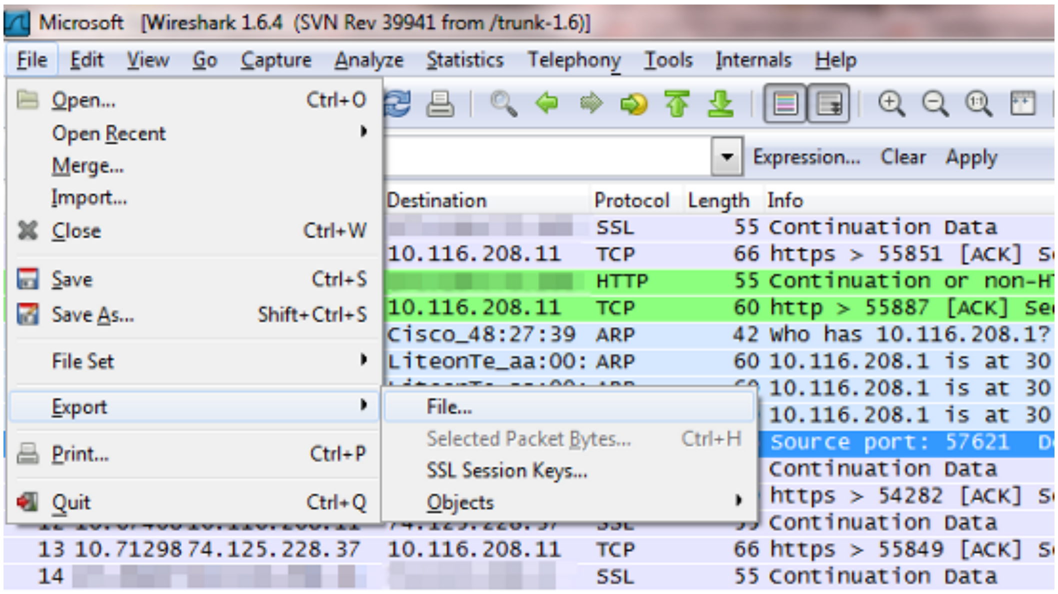
Data Analysis

1. Once the captures have been completed, use TFTP to export them to a PC.
2. Open the captures with a network protocol analyzer (such as Wireshark).
3. If QoS marking was used, filter out the respective packets.

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

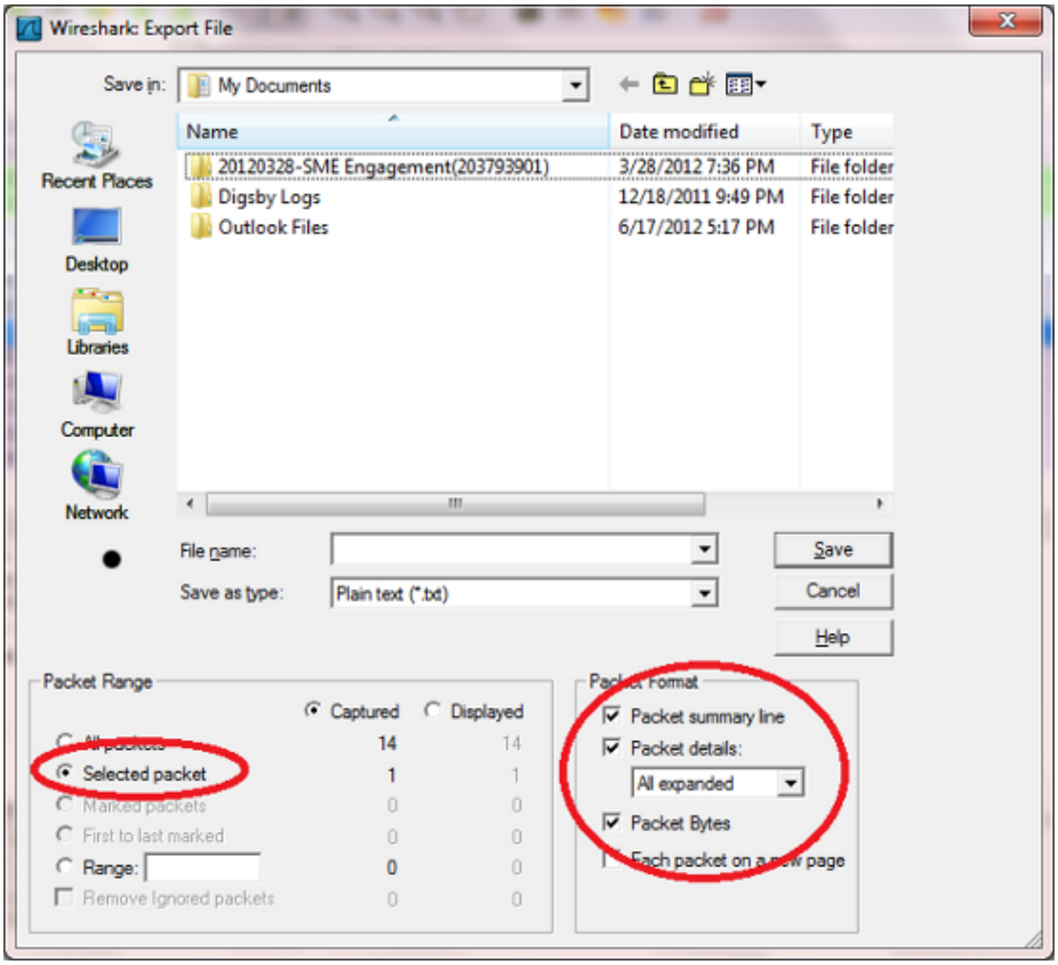
   "0x08" is specific for the DSCP value AF21. If a different DSCP value is used, the correct value can be obtained from the packet capture itself or from the list of DSCP values conversion chart. Refer to DSCP and Precedence Values for more information.

4. Identify the dropped ping on the captures from the sender, and locate that packet on captures on both the receiver side and the sender side.
5. Export that packet from both captures as shown in this image:



Microsoft [Wireshark 1.6.4 (SVN Rev 39941 from /trunk-1.6)]

```
Destination     Protocol Length Info
                SSL       55 Continuation Data
10.116.208.11   TCP       66 https > 55851 [ACK] S
                HTTP      55 Continuation or non-H
10.116.208.11   TCP       60 http > 55887 [ACK] Se
Cisco_48:27:39  ARP       42 Who has 10.116.208.1?
LiteonTe_aa:00: ARP       60 10.116.208.1 is at 30
                             10.116.208.1 is at 30
                             10.116.208.1 is at 30
                             Source port: 57621  D
                             Continuation Data
                             https > 54282 [ACK] S
                             Continuation Data
13 10.71298 74.125.228.37  10.116.208.11  TCP  66 https > 55849 [ACK] S
14                                        SSL  55 Continuation Data
```



Wireshark: Export File

Save in: My Documents

Name	Date modified	Type
20120328-SME Engagement(203793901)	3/28/2012 7:36 PM	File folder
Digsby Logs	12/18/2011 9:49 PM	File folder
Outlook Files	6/17/2012 5:17 PM	File folder

File name:
Save as type: Plain text (*.txt)

Packet Range

	Captured	Displayed
All packets	14	14
[✓ Selected packet]	1	1
Marked packets	0	0
First to last marked	0	0
Range:	0	0
Remove Ignored packets	0	0

Packet Format
- [✓] Packet summary line
- [✓] Packet details: All expanded
- [✓] Packet Bytes
- [ ] Each packet on a new page

6. Conduct a binary comparison of the two. If they are identical, then there were no errors in transit and the Cisco IOS either threw a false negative on the receiving end or used the wrong key on the sender end. In either case, the issue is a Cisco IOS bug. If the packets are different, then the packets were tampered with in transmit.

Here is the packet as it left the crypto engine on the FC:

```
*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.[!.
```

Here is the same packet as it was received on the peer:

```
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  ................
```

At this point, it is most likely an ISP problem, and that group should be involved in the troubleshooting.

Common Problems

- Cisco bug ID CSCed87408 describes a hardware issue with the crypto engine on the 83xs where random outgoing packets are corrupted during encryption, which leads to authentication errors (in cases where authentication is used) and packet drops on the receiving end. It is important to realize that you will not see these errors on the 83x itself, but on the receiving device.
- Sometimes routers that run old code show this error. You can upgrade to the more recent code versions such as 15.1(4) M4 to resolve the issue.
- In order to verify if the problem is a hardware or software issue, disable hardware encryption. If the log messages continue, it is a software issue. If not, then an RMA should resolve the problem. Remember that if you disable hardware encryption, it can cause severe network degradation for heavily loaded VPN tunnels. Therefore, Cisco recommends you attempt the procedures described in this document during a maintenance window.

Related Information

- ***Technical Support & Documentation − Cisco Systems***