pfSense - Bug #7969
md5 bgp sessions fail in 2.4.0
10/18/2017 10:09 PM - Andrew Dul

<table>
<thead>
<tr>
<th>Status:</th>
<th>Resolved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Luiz Souza</td>
</tr>
<tr>
<td>Category:</td>
<td>Routing</td>
</tr>
<tr>
<td>Start date:</td>
<td>10/18/2017</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.4.2</td>
</tr>
<tr>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Primail:</td>
<td></td>
</tr>
<tr>
<td>Due date:</td>
<td>10/18/2017</td>
</tr>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.4.2</td>
</tr>
<tr>
<td>Affected Version:</td>
<td>2.4</td>
</tr>
<tr>
<td>Est. time:</td>
<td>0.00 hour</td>
</tr>
</tbody>
</table>

**Description**

Upgraded to 2.4.0 from 2.3.4 and my bgp sessions which were secured via TCP md5 configurations in openbgpd & the new frr routing package.

My routers which are upstream show the following error message
%TCP-6-BADAUTH: Invalid MD5 digest from [peerA]:xxx to [peerB]:179

I reverted back to 2.3.4 and was able to successfully make the sessions work with the same configuration under both openbgpd & frr.

**History**

#1 - 10/20/2017 09:26 PM - Jim Thompson
- Assignee set to Jim Pingle
- Target version set to 2.4.2

#2 - 10/20/2017 09:27 PM - Jim Thompson
- Category changed from IPsec to Routing

#3 - 10/21/2017 11:32 AM - Jim Pingle
Do you have "BSD Crypto Device" selected under System > Advanced, Misc tab, for Cryptographic Hardware? If not, select it there and try again.

That module is required for TCP_SIGNATURE to function.

If that works I can either add some warning text to Quagga and FRR or force it to load when that is enabled.

#4 - 10/23/2017 05:49 PM - Andrew Dul
- File ScreenHunter_01 Oct. 23 13.27.jpg added
- File sw1.txt added
- File config-pfSense.localdomain-20171023202849.xml added

I was able to reproduce this on pfsense 2.3.4 vs 2.4.0 w/ fresh installs, running in virtual box w/ an Arista vEOS VM as the other bgp neighbor.

I've attached the basic config that I used for both pfsense & arista vEOS to test.

I also checked the "BSD Crypto Device" option. It was enabled by default when I checked on 2.4.0.

#5 - 10/24/2017 03:25 PM - Terry Zink
Currently seeing this same issue. Updated to 2.4.0 from 2.3.x and my AWS Direct Connect sessions broke. AWS Support notes I am not sending the MD5 key with my tcp packets.

I do have BSD Crypto Device enabled, and this does not make any difference.

edit: Also tested 2.4.1, same issue persists.

#6 - 10/25/2017 03:02 AM - Jim Thompson
[https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=219453](https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=219453)
Can someone, please, provide the output of 'ifconfig -v' of affected interfaces, 'kidstat' and 'netstat -sp tcp'.

Sure thing. Files attached (ip info scrubbed).

Can you also get the output of setkey -D and setkey -DP?

Attached.

Note in my case: both interfaces/BGP peers have the same key. (Intended)

Here is what I see on the lab setup. Both 2.3.4 and 2.4.0.

Definitely seems like it's deeper than the routing daemons. I tried the same config with FRR on 2.3.x and 2.4.x and on 2.4.x, the setkey entry never gets any traffic. Almost as if it doesn't find it interesting, though it does match.

2.3.5:

```
s.s.s.s  d.d.d.d
  tcp mode=any spi=4096(0x00001000) reqid=0(0x00000000)
  A: tcp-md5 61626331 3233
  seq=0x00000000 replay=0 flags=0x0000040 state=mature
  diff: 6(s) hard: 0(s) soft: 0(s)
  last: Oct 25 13:54:26 2017  hard: 0(s) soft: 0(s)
  current: 0(bytes) hard: 0(bytes) soft: 0(bytes)
  allocated: 3 hard: 0 soft: 0
  sadbh_seq=2 pid=37461 refcnt=1
```

2.4.2:

```
s.s.s.s  d.d.d.d
  tcp mode=any spi=4096(0x00001000) reqid=0(0x00000000)
```

06/26/2022
A: tcp-md5 61626331 3233
seq=0x00000000 replay=0 flags=0x00000040 state=mature
diff: 1294(s) hard: 0(s) soft: 0(s)
last: hard: 0(s) soft: 0(s)
current: 0(bytes) hard: 0(bytes) soft: 0(bytes)
allocated: 0 hard: 0 soft: 0
sadb_seq=2 pid=16507 refcnt=1

: netstat -sp tcp | grep sign
0 packets with matching signature received
0 packets with bad signature received
0 times failed to make signature due to no SA
0 times unexpected signature received
0 times no signature provided by segment

#13 - 11/02/2017 10:40 AM - Terry Zink
Downgraded my device back to 2.3.4 after taking the trip out to the DC. Working fine now. Definitely 2.4.x related.

#14 - 11/02/2017 03:56 PM - Tim Economides
All - I did some digging and found that when I built MD5 support into Quagga (code which was subsequently used in developing the FRR package) that it was only set to build the setkey config file using SPI 0x1000 for source-dest traffic, and another line needs to be added for SPI 0x1001 dest-source traffic. I've got an alpha test working in my lab environment. Once I've tested further, I'll submit an update to both packages in github.

#15 - 11/03/2017 09:21 AM - Tim Economides
Tim Economides wrote:
All - I did some digging and found that when I built MD5 support into Quagga (code which was subsequently used in developing the FRR package) that it was only set to build the setkey config file using SPI 0x1000 for source-dest traffic, and another line needs to be added for SPI 0x1001 dest-source traffic. I've got an alpha test working in my lab environment. Once I've tested further, I'll submit an update to both packages in github.

The code in quagga_ospf.inc changes from:

```perl
foreach ($config['installedpackages'][0]['config'][0]['row'] as $bgpdpw) {
    if (($bgpdpw['bgpdsourceaddr'] != "") && ($bgpdpw['bgpdpeeraddr'] != ") && ($bgpdpw['bgpdmd5pw'])) != ") {
        $bgpdaddmd5file .= "add \{bgpdwpw['bgpdsourceaddr']\} \{bgpdwpw['bgpdpeeraddr']\} tcp 0x1000 -A tcp-md5 \"{$bgpdwpw['bgpdmd5pw']}\" ;n;
        $bgpddelmd5file .= "delete \{bgpdwpw['bgpdsourceaddr']\} \{bgpdwpw['bgpdpeeraddr']\} tcp 0x1000 ;\n"
    }
}
```

to

```perl
foreach ($config['installedpackages'][0]['config'][0]['row'] as $bgpdpw) {
    if (($bgpdpw['bgpdsourceaddr'] != "") && ($bgpdpw['bgpdpeeraddr'] != "") && ($bgpdpw['bgpdmd5pw'])) != ") {
        $bgpdaddmd5file .= "add \{bgpdwpw['bgpdsourceaddr']\} \{bgpdwpw['bgpdpeeraddr']\} tcp 0x1000 -A tcp-md5 \"{$bgpdwpw['bgpdmd5pw']}\" ;n;
        $bgpdaddmd5file .= "add \{bgpdwpw['bgpdpeeraddr']\} \{bgpdwpw['bgpdsourceaddr']\} tcp 0x1001 -A tcp-md5 \"{$bgpdwpw['bgpdmd5pw']}\" ;n;
        $bgpddelmd5file .= "delete \{bgpdwpw['bgpdsourceaddr']\} \{bgpdwpw['bgpdpeeraddr']\} tcp 0x1000 ;\n"
    }
}
```

Basically, the code should generate a bgpdaddmd5pw.conf that looks like (where x.x.x.x is the local router and y.y.y.y is the peer and the password is "p@ssw0rd"):

```
add x.x.x.x y.y.y.y tcp 0x1000 -A tcp-md5 "p@ssw0rd"
add y.y.y.x x.x.x.x tcp 0x1001 -A tcp-md5 "p@ssw0rd"
```

and bgpddeled5pw.conf is generated as:

```bash
06/26/2022 3/8
```
#16 - 11/03/2017 10:57 AM - Jim Pingle

Those changes do seem to be corroborated by the setkey(8) man page for FreeBSD 11.1, but they do not appear to actually help. They were omitted in the past because FreeBSD was not capable of validating TCP MD5 signatures, only setting them outbound. That may have changed in 11.1 with the new IPsec stack import.

Even with the additional SA entry, however, I still see no packets hitting the SA in either direction, nor does it appear to have signed anything. tcpdump does not show a signature in the packets. Does it actually appear to be working for you with the additional SA?

#17 - 11/03/2017 01:10 PM - Tim Economides

Jim Pingle wrote:

Those changes do seem to be corroborated by the setkey(8) man page for FreeBSD 11.1, but they do not appear to actually help. They were omitted in the past because FreeBSD was not capable of validating TCP MD5 signatures, only setting them outbound. That may have changed in 11.1 with the new IPsec stack import.

Even with the additional SA entry, however, I still see no packets hitting the SA in either direction, nor does it appear to have signed anything. tcpdump does not show a signature in the packets. Does it actually appear to be working for you with the additional SA?

I see the SA is applied and incrementing its traffic counter in both directions. Additionally, a tcpdump on port 179 shows only md5 signed traffic between my routers, so I'm thinking this is working properly. I have zero issues with Cisco routers that are sending and receiving md5 signed data. If the md5 password is not specified when initiating the tcpdump, a "md5 shared secret not supplied with -M, can't check" error is produced. Providing the specified password results in the traffic being viewed, as expected.

Let me see if I can get some data out of my lab to show you - it's in a bubble that makes it a bit difficult.

#18 - 11/03/2017 01:37 PM - Jim Pingle

I'd be surprised if it was actually working due to that change alone. Maybe you changed something else unrelated to just the second SA. It's nice to have that second SA, but not necessary.

I tried replicating this on a few different environments:

- pfSense 2.4.2 snapshots - fails but it is not yet clear why. The kernel has IPSEC and TCP_SIGNATURE (but no IPSEC_SUPPORT as it shouldn't be needed with both compiled in the kernel), the SA is present, traffic should be matching but doesn't appear to be, even with an SA for each direction.
- Binary install of FreeBSD 11.1 - fails because you can't kldload tcpmd5 without IPSEC_SUPPORT in the kernel, it only has IPSEC
- Source upgraded FreeBSD stable/11 system - worked after kldload tcpmd5 with just the one SA for 0x1000 (It has IPSEC and IPSEC_SUPPORT in the kernel), but it's not clear if it works because it's plain FreeBSD or if it works because it's stable/11.

I don't have a FreeBSD 11.1-RELEASE box handy that has a custom kernel with both IPSEC and TCP_SIGNATURE built-in to compare against though, I'll try to get one setup.

#19 - 11/03/2017 02:00 PM - Tim Economides

Jim Pingle wrote:

I'd be surprised if it was actually working due to that change alone. Maybe you changed something else unrelated to just the second SA. It's nice to have that second SA, but not necessary.

I tried replicating this on a few different environments:

- pfSense 2.4.2 snapshots - fails but it is not yet clear why. The kernel has IPSEC and TCP_SIGNATURE (but no IPSEC_SUPPORT as it shouldn't be needed with both compiled in the kernel), the SA is present, traffic should be matching but doesn't appear to be, even with an SA for each direction.
- Binary install of FreeBSD 11.1 - fails because you can't kldload tcpmd5 without IPSEC_SUPPORT in the kernel, it only has IPSEC
- Source upgraded FreeBSD stable/11 system - worked after kldload tcpmd5 with just the one SA for 0x1000 (It has IPSEC and IPSEC_SUPPORT in the kernel), but it's not clear if it works because it's plain FreeBSD or if it works because it's stable/11.

I don't have a FreeBSD 11.1-RELEASE box handy that has a custom kernel with both IPSEC and TCP_SIGNATURE built-in to compare against though, I'll try to get one setup.

Very interesting. I'm running pfSense 2.4.1-RELEASE, and tcpmd5 is loaded properly. Just finished a full packet capture and associated analysis to verify it's working, and unless I'm blind, it appears to be. setkey -D shows traffic incrementing in both directions for each defined SPI, I'm not sure what else to say.

Here's my setkey output from each peer, note the "current: ######(bytes)" field. This increments both ways on both hosts.
[2.4.1-RELEASE] [admin@XXX-pfSense-01a.quad.org]/root: setkey -D
xxx.yyy.128.43 xxx.yyy.128.42
tcp mode=any spi=4097(0x00001001) reqid=0(0x00000000)
At: tcp-md5 70617373 776f7264
seq=0x00000000 replay=0x00000040 state=mature
diff: 2115(s) hard: 0(s) soft: 0(s)
last: Nov 3 14:39:06 2017 hard: 0(s) soft: 0(s)
current: 12373(bytes) hard: 0(bytes) soft: 0(bytes)
allocated: 149 hard: 0 soft: 0
sadb_seq=1 pid=13310 refcnt=1

[2.4.1-RELEASE] [admin@XXX-pfSense-01b.quad.org]/root: setkey -D
xxx.yyy.128.42 xxx.yyy.128.43
tcp mode=any spi=4096(0x00001000) reqid=0(0x00000000)
At: tcp-md5 70617373 776f7264
seq=0x00000000 replay=0 flags=0x00000040 state=mature
diff: 2115(s) hard: 0(s) soft: 0(s)
last: Nov 3 14:39:06 2017 hard: 0(s) soft: 0(s)
current: 16766(bytes) hard: 0(bytes) soft: 0(bytes)
allocated: 205 hard: 0 soft: 0
sadb_seq=0 pid=13310 refcnt=1

Could be quagga vs frr, I am testing with frr. I'm still not convinced the second SA is doing anything to help the situation though.

#21 - 11/03/2017 02:41 PM - Tim Economides

Jim Pingle wrote:

Could be quagga vs frr, I am testing with frr. I'm still not convinced the second SA is doing anything to help the situation though.

Perhaps; I haven't gotten around to working on this on FRR. Considering I was completely unable to peer previously when using md5 before adding the second SA according to the setkey manpage for FreeBSD 11.x, and I'm seeing the SA apply, traffic increment, and tcpdumps indicating signed traffic, I'm pretty certain this is working on Quagga at least. I'll revisit it with FRR next week and if I make progress let's compare notes.

#22 - 11/04/2017 01:03 PM - Terry Zink

Worth noting I have been seeing all of this with openbgpd, so it would be strange if it was specific to routing daemons.

#23 - 11/07/2017 02:58 PM - Tim Economides

Jim Pingle wrote:

Could be quagga vs frr, I am testing with frr. I'm still not convinced the second SA is doing anything to help the situation though.

I have this working successfully with FRR using a similar modification to frr.inc (beginning at line 424) as used in quagga_ospfd.inc:

foreach ($peers as $peer) {

06/26/2022
As verified with Quagga, SA stats are incrementing in both directions and pcaps verify signed packets.

#24 - 11/08/2017 12:51 PM - Jim Pingle
I just pushed a change to FRR to allow the user to manually choose whether or not they want to use setkey entries for both directions.

Also on a current snapshot I am now seeing working TCP MD5 from FRR, in one or both directions. Luiz made a kernel option change a few days ago that appears to have helped.

OpenBGPD still does not appear to work, however. It adds inbound and outbound setkey entries, The outbound entry looks OK and shows traffic but the far side doesn't like what it's sending. The inbound entry has the wrong spi and there is no way in the configuration to change that for TCP MD5 that I see. It's possible that it may be OK but not with the setup I have, since I did not have OpenBGPD working with TCP MD5 previously. It looks like it should be, but perhaps a different peer may work better than my limited test.

#25 - 11/08/2017 01:33 PM - Tim Economides
Jim Pingle wrote:

I just pushed a change to FRR to allow the user to manually choose whether or not they want to use setkey entries for both directions.

Also on a current snapshot I am now seeing working TCP MD5 from FRR, in one or both directions. Luiz made a kernel option change a few days ago that appears to have helped.

OpenBGPD still does not appear to work, however. It adds inbound and outbound setkey entries, The outbound entry looks OK and shows traffic but the far side doesn't like what it's sending. The inbound entry has the wrong spi and there is no way in the configuration to change that for TCP MD5 that I see. It's possible that it may be OK but not with the setup I have, since I did not have OpenBGPD working with TCP MD5 previously. It looks like it should be, but perhaps a different peer may work better than my limited test.

Thanks Jim; looks good though not for raw configs, can we add a similar bidirectional dropdown in the raw config page as well?

Another note about the raw config page - you can add as many md5 entries as you want, but only the first gets saved.

#26 - 11/08/2017 02:40 PM - Jim Pingle
I added the flag to the raw config page. Unfortunately, fixing the other bug meant I had to rename the fields so the old values will be missing on upgrade, but now you can store as many rows as you need. There wasn't a good way to 'upgrade' the code in-place the way the package was designed. I'll keep poking at that but having it working is better than leaving it broken. That should probably be on a different ticket, however.

#27 - 11/08/2017 03:03 PM - Tim Economides
Jim Pingle wrote:

I added the flag to the raw config page. Unfortunately, fixing the other bug meant I had to rename the fields so the old values will be missing on upgrade, but now you can store as many rows as you need. There wasn't a good way to 'upgrade' the code in-place the way the package was designed. I'll keep poking at that but having it working is better than leaving it broken. That should probably be on a different ticket, however.

Tested and verified; thanks for the quick fix.

Somewhat related issue I've first noticed while working with Quagga and FRR - When working with "rowhelper" fields in the pkg xml files, the last entered item on any given row is not saved. Since this is universal to all packages using that field type i suspect it has more to do with code in the pkg_edit.php file than anything else, have you observed this before?

#28 - 11/08/2017 03:10 PM - Jim Pingle
Tim Economides wrote:

Somewhat related issue I've first noticed while working with Quagga and FRR - When working with "rowhelper" fields in the pkg xml files, the last entered item on any given row is not saved. Since this is universal to all packages using that field type i suspect it has more to do with code in the pkg_edit.php file than anything else, have you observed this before?

That's very off topic for this ticket, but no, they always save for me. If you have more questions that are not directly related to MD5+BGP, start a thread on the forum or reddit and it can be discussed there.

#29 - 11/09/2017 09:14 AM - Jim Pingle
- Status changed from New to Resolved
Anything at the OS level appears to be fine now. I am able to establish a BGP peering with TCP MD5 and the latest FRR between two pfSense VMs directly connected. I can also establish a peering between FRR and OpenBGPD with an MD5 password.

#30 - 12/08/2017 12:36 PM - Andrew Dul
I downloaded the new 2.4.2 and tried to get this working and still was unable to make it work.

The "Type of Password" drop down is a bit confusing. I've used the same password with FRR & my BGP neighbor and selected "FRR and setkey Bidirectional" but that didn't work. The test switch Arista vEOS still reports

Bgp: %TCP-6-BADAUTH: No MD5 digest from 192.168.1.1(33577) to 192.168.1.50(179)

Are there other changes that I need to make?

#31 - 12/08/2017 12:56 PM - Andrew Dul
- File configs.txt added

Attaching config files from /var/etc/frr

#32 - 03/21/2018 02:48 PM - Matthew Fields
I am using OpenBGPD on 2.3.5 and am peering using an MD5 password to a Cisco device, when I upgraded to 2.4.2, the MD5 password is not getting passed through at all or at least incorrectly (according to the receiving end).

Is there anything above that would be helpful to get OpenBGPD working with MD5 in 2.4.x?

#33 - 03/31/2018 11:52 AM - Anonymous
I recently upgraded some systems from 2.3.5 to 2.4.3 and found that FRR BGP MD5 support is now broken. When the outgoing interface is physical / LAGG it was sufficient to enable hardware checksum support to fix the issue. When the outgoing interface is an OpenVPN tunnel there is no such option, so BGP MD5 support is still broken.

A new patch in [https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=223835](https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=223835) seems to fix this problem for any interface type by removing the hardware checksum requirement. Is it possible to pull in that patch for the next release?

#34 - 04/03/2018 03:35 PM - Matthew Fields
bkraptor, where is the "Enable Hardware Checksum Support" listed at? I could not seem to find it except for a checkbox to DISABLE hardware checksum.

Thanks!

#35 - 04/03/2018 03:43 PM - Andrew Dul
Can someone reopen this bug, it certainly doesn't seem like it has been resolved based on multiple people testing

#36 - 04/03/2018 03:47 PM - Anonymous
I have already opened #8407 for this issue, so feel free to continue the conversation there.

@Matthew Fields: that's the exact checkbox that triggers the enable/disable behavior I was referring to. The checkbox needs to be unticked for hardware checksum support to be enabled.

#37 - 04/05/2018 11:08 AM - Matthew Fields
bkraptor - wrote:

I have already opened #8407 for this issue, so feel free to continue the conversation there.

@Matthew Fields: that's the exact checkbox that triggers the enable/disable behavior I was referring to. The checkbox needs to be unticked for hardware checksum support to be enabled.

Okay, mine is enabled by default, however, it still has the issue with the remote side not receiving the password (MD5). I have for the time being reverted to 2.3.5.1 and will stay there until 2.4.x is fixed.

Files

- sw1.txt 616 Bytes 10/23/2017 Andrew Dul
- config-pfSense.localdomain-20171023202849.xml 19 KB 10/23/2017 Andrew Dul
- ScreenHunter_01 Oct. 23 13.27.jpg 266 KB 10/23/2017 Andrew Dul
- ifconfig.txt 2.01 KB 10/25/2017 Terry Zink
- netstat.txt 7.52 KB 10/25/2017 Terry Zink

06/26/2022