batman-adv - Bug #356

TT: XOR'ing CRC results unsafe

05/10/2018 02:31 PM - Linus Lüssing

Status:	New	**Start date:**	05/10/2018
Priority:	Normal	**Due date:**	
Assignee:	batman-adv developers	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Currently, the custom CRC caluclation for TT in batman-adv works as follows:

- Compute the CRC for each entry, including vid and TT sync flags.
- Then XOR all resulting CRCs

However, while playing with injecting flags to multicast entries we now noticed that XOR'ing CRCs seems to possibly have easy collision issues:

```
root@Linus-Debian:/mnt/batman-adv-t_x# batctl tg
[B.A.T.M.A.N. adv 2018.1-10-gc0c5f610, MainIF/MAC: ens3/02:32:64:a4:39:c1 (bat0/0a:f0:8e:ca:5e:82
BATMAN_IV)]
   Client              VID Flags Last ttvn     Via         ttvn  (CRC       )
 * 0e:b3:20:0c:05:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
 * 33:33:ff:0c:05:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
   01:00:5e:00:00:01   -1 [....] (  2) 02:01:64:a4:39:c4 (  2) (0xe0acdb32)
   01:00:5e:00:00:01   -1 [....] (  2) 02:84:64:a4:39:c3 (  2) (0xfefb94e8)
 * 01:00:5e:00:00:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
 * 33:33:ff:c8:3a:24   -1 [....] (  2) 02:84:64:a4:39:c3 (  2) (0xfefb94e8)
 * 4a:1d:5e:5c:79:c9   -1 [....] (  2) 02:01:64:a4:39:c4 (  2) (0xe0acdb32)
 * 33:33:ff:5c:79:c9   -1 [....] (  2) 02:01:64:a4:39:c4 (  2) (0xe0acdb32)
 * fe:04:73:c8:3a:24   -1 [....] (  2) 02:84:64:a4:39:c3 (  2) (0xfefb94e8)
   33:33:00:00:00:01   -1 [....] (  2) 02:01:64:a4:39:c4 (  2) (0xe0acdb32)
   33:33:00:00:00:01   -1 [....] (  2) 02:84:64:a4:39:c3 (  2) (0xfefb94e8)
 * 33:33:00:00:00:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
root@Linus-Debian:/mnt/batman-adv-t_x# batctl tg
[B.A.T.M.A.N. adv 2018.1-10-gc0c5f610, MainIF/MAC: ens3/02:32:64:a4:39:c1 (bat0/0a:f0:8e:ca:5e:82
BATMAN_IV)]
   Client              VID Flags Last ttvn     Via         ttvn  (CRC       )
 * 0e:b3:20:0c:05:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
 * 33:33:ff:0c:05:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
   01:00:5e:00:00:01   -1 [.W..] (  4) 02:01:64:a4:39:c4 (  4) (0x7cf72194)
   01:00:5e:00:00:01   -1 [....] (  2) 02:84:64:a4:39:c3 (  2) (0xfefb94e8)
 * 01:00:5e:00:00:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
 * 33:33:ff:c8:3a:24   -1 [....] (  2) 02:84:64:a4:39:c3 (  2) (0xfefb94e8)
 * 4a:1d:5e:5c:79:c9   -1 [....] (  4) 02:01:64:a4:39:c4 (  4) (0x7cf72194)
 * 33:33:ff:5c:79:c9   -1 [.W..] (  4) 02:01:64:a4:39:c4 (  4) (0x7cf72194)
 * fe:04:73:c8:3a:24   -1 [....] (  2) 02:84:64:a4:39:c3 (  2) (0xfefb94e8)
   33:33:00:00:00:01   -1 [.W..] (  4) 02:01:64:a4:39:c4 (  4) (0x7cf72194)
   33:33:00:00:00:01   -1 [....] (  2) 02:84:64:a4:39:c3 (  2) (0xfefb94e8)
 * 33:33:00:00:00:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
root@Linus-Debian:/mnt/batman-adv-t_x# batctl tg
[B.A.T.M.A.N. adv 2018.1-10-gc0c5f610, MainIF/MAC: ens3/02:32:64:a4:39:c1 (bat0/0a:f0:8e:ca:5e:82
BATMAN_IV)]
   Client              VID Flags Last ttvn     Via         ttvn  (CRC       )
 * 0e:b3:20:0c:05:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
 * 33:33:ff:0c:05:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
   01:00:5e:00:00:01   -1 [.W..] (  4) 02:84:64:a4:39:c3 (  4) (0xfefb94e8)
   01:00:5e:00:00:01   -1 [.W..] (  4) 02:01:64:a4:39:c4 (  4) (0x7cf72194)
 * 01:00:5e:00:00:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
 * 33:33:ff:c8:3a:24   -1 [....] (  4) 02:84:64:a4:39:c3 (  4) (0xfefb94e8)
 * 4a:1d:5e:5c:79:c9   -1 [....] (  4) 02:01:64:a4:39:c4 (  4) (0x7cf72194)
 * 33:33:ff:5c:79:c9   -1 [.W..] (  4) 02:01:64:a4:39:c4 (  4) (0x7cf72194)
```

```
 * fe:04:73:c8:3a:24   -1 [....] (  4) 02:84:64:a4:39:c3 (  4) (0xfefb94e8)
   33:33:00:00:00:01   -1 [.W..] (  4) 02:84:64:a4:39:c3 (  4) (0xfefb94e8)
   33:33:00:00:00:01   -1 [.W..] (  4) 02:01:64:a4:39:c4 (  4) (0x7cf72194)
 * 33:33:00:00:00:01   -1 [....] (  2) 02:05:64:a4:39:c2 (  2) (0xdbb13619)
root@Linus-Debian:/mnt/batman-adv-t_x#
```

When the 'W' flag was injected on node ..:c4 on three of its entries, as expected the result was a new CRC (before: 0xe0acdb32, after: 0x7cf72194).

However, when the 'W' flag was injected on node ..:c3 and only on two of its entries, the CRC stayed the same (0xfefb94e8).

It seems that xor'ing two CRC results which both had exactly the same one bit flipped nullifies the change introduced by this bit.

Sample code to verify:

```
#!/usr/bin/python3

import binascii

a=b"1234"
b=b"4305"

crca0=binascii.crc32(a, 0x0)
crcb0=binascii.crc32(b, 0x0)
xor0=crca0 ^ crcb0

print('CRC({:s}, 0x0) ^ CRC({:s}, 0x0) = {:#010x} ^ {:#010x} = {:#010x}'.format(str(a), str(b), cr
ca0, crcb0, xor0))

crca1=binascii.crc32(a, 0x1)
crcb1=binascii.crc32(b, 0x1)
xor1=crca1 ^ crcb1

print('CRC({:s}, 0x1) ^ CRC({:s}, 0x1) = {:#010x} ^ {:#010x} = {:#010x}'.format(str(a), str(b), cr
ca1, crcb1, xor1))
```

Output:

```
CRC(b'1234', 0x0) ^ CRC(b'4305', 0x0) = 0x9be3e0a3 ^ 0xf1d519f3 = 0x6a36f950
CRC(b'1234', 0x1) ^ CRC(b'4305', 0x1) = 0x235f87c6 ^ 0x49697e96 = 0x6a36f950
```

The reason for XOR'ing instead of only CRC'ing back then seems to have been to be able to be independent of the order of TT entries. Note that any fix by changing the checksumming method batman-adv uses for TT will likely not be backwards compatible.

History

#1 - 05/10/2018 02:42 PM - Antonio Quartulli

after searching online, it seems that a widely strategy to compute a hash of a set (=unordered collection of elements) consists in:

- compute the hash of each elements
- sort the computed hashes lexicographically
- concatenate the hashes in a "non-ambiguous" way to form a new string
- hash the resulting string.

The hash obtained after the last step is the signature/hash of the set.

Unfortunately this cannot be done on-the-fly while traversing the TT table, but requires to store all the hashes of the various elements first.

There seems to be a paper describing how to compute the hash of a set with constant memory[1], but I am not sure there is an implementation

available to take inspiration from.

[1]https://people.csail.mit.edu/devadas/pubs/mhashes.pdf