THE GEEK STUFF
Linux | DB | Open Source | Web

≡ Menu

- Home
- Free eBook
- Start Here
- Contact
- About

How to Calculate IP Header Checksum (With an Example)

by Himanshu Arora on May 17, 2012

If you have ever tried to understand the TCP/IP protocols then you would have definitely stumbled upon the checksum field that is the part of protocol headers like TCP, IP etc.

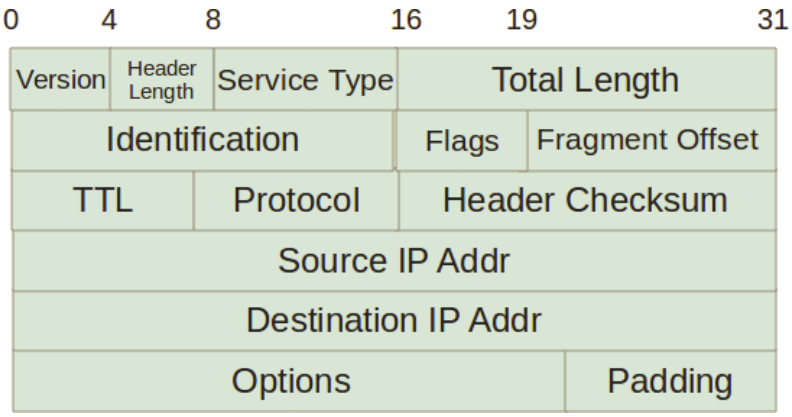
Have you ever given a thought about things like what exactly is checksum, why is it used and how it is calculated. Well, in this article we will have a brief discussion on the concept of checksum and then we will go into details of how checksum is calculated.

What is Checksum?

A check sum is basically a value that is computed from data packet to check its integrity. Through integrity, we mean a check on whether the data received is error free or not. This is because while traveling on network a data packet can become corrupt and there has to be a way at the receiving end to know that data is corrupted or not. This is the reason the checksum field is added to the header. At the source side, the checksum is calculated and set in header as a field. At the destination side, the checksum is again calculated and crosschecked with the existing checksum value in header to see if the data packet is OK or not.

IP header checksum

IP header checksum is calculated over IP header only as the data that generally follows the IP header (like ICMP, TCP etc) have their own checksums. Now, to calculate the IP header algorithm one must know the basic header structure of IP protocol. So here is a basic format of how IP header looks like :



NOTE: To have a good understanding of the IP header fields, refer to: IP Protocol Header Fundamentals

So, as far as the algorithm goes, IP header checksum is : ***16 bit one's complement of the one's complement sum of all 16 bit words in the header***

This means that if we divide the IP header is 16 bit words and sum each of them up and then finally do a one's compliment of the sum then the value generated out of this operation would be the checksum.

Now, the above is done at the source side which is sending the data packet. At the destination side which receives the data packet replaces the checksum value in the header with all zeros and then calculates the checksum based on the same algorithm as mentioned above. After a checksum value is obtained then this value is compared with the value that came in the header. This comparison decides whether the IP header is fine or corrupted.

IP Header Checksum Example

Since now we have enough theoretical knowledge on IP header checksum, lets take an IP header and actually try this algorithm out.

Here is a IP header from an IP packet received at destination :

```
4500 003c 1c46 4000 4006 b1e6 ac10 0a63 ac10 0a0c
```

Lets first map these values with the header

- '45' corresponds to the first two fields in the header ie '4' corresponds to the IP version and '5' corresponds to the header length. Since header length is described in 4 byte words so actual header length comes out to be 5×4=20 bytes.
- '00' corresponds to TOS or the type of service. This value of TOS indicated normal operation.
- '003c' corresponds to total length field of IP header. So in this case the total length of IP packet is 60.
- '1c46′ corresponds to the identification field.
- '4000' can be divided into two bytes. These two bytes (divided into 3 bits and 13 bits respectively) correspond to the flags and fragment offset of IP header fields.
- '4006' can be divided into '40' and '06'. The first byte '40' corresponds to the TTL field and the byte '06' corresponds to the protocol field of the IP header. '06' indicates that the protocol is TCP.
- **'be16′ corresponds to the checksum which is set at the source end (which sent the packet)**. Please note that as already discussed this field will be set to zero while computing the checksum at destination end.
- The next set of bytes 'ac10′ and '0a0c' correspond to the source IP address and the destination IP address in the IP header.

So now we have a basic idea as to what these fields map to in IP header. Lets convert all these values in binary :

```
4500 -> 0100010100000000
003c -> 0000000000111100
1c46 -> 0001110001000110
4000 -> 0100000000000000
4006 -> 0100000000000110
0000 -> 0000000000000000 // Note that the checksum is set to zero since we are computing checksum at destination end
ac10 -> 1010110000010000
0a63 -> 0000101001100011
ac10 -> 1010110000010000
0a0c -> 0000101000001100
```

Now lets add these binary values one by one :

```
4500 -> 0100010100000000
003c -> 0000000000111100
453C -> 0100010100111100  /// First result

453C -> 0100010100111100  // First result plus next 16-bit word.
1c46 -> 0001110001000110
6182 -> 0110000110000010 // Second result.

6182 -> 0110000110000010 // Second result plus next 16-bit word.
4000 -> 0100000000000000
A182 -> 1010000110000010 // Third result.

A182 -> 1010000110000010 // Third result plus next 16-bit word.
4006 -> 0100000000000110
E188 -> 1110000110001000 // Fourth result.

E188 -> 1110000110001000 // Fourth result plus next 16-bit word.
AC10 -> 1010110000010000
18D98 -> 11000110110011000 // One odd bit (carry),  add that odd bit to the result as we need to keep the checksum in 16 bits.

18D98 -> 11000110110011000
8D99 -> 1000110110011001 // Fifth result

8D99 -> 1000110110011001 // Fifth result plus next 16-bit word.
0A63 -> 0000101001100011
97FC -> 1001011111111100 // Sixth result

97FC -> 1001011111111100  // Sixth result plus next 16-bit word.
AC10 -> 1010110000010000
1440C -> 10100010000001100 // Again a carry, so we add it (as done before)

1440C -> 10100010000001100
440D -> 0100010000001101 // This is seventh result

440D -> 0100010000001101 //Seventh result plus next 16-bit word
0A0C -> 0000101000001100
4E19 -> 0100111000011001 // Final result.
```

So now 0100111000011001 is our final result of summing up all the 16 bit words in the header. As a last step we just need to do a one's compliment of it to obtain the checksum.

```
4E19 -> 0100111000011001
B1E6 ->1011000111100110 // CHECKSUM
```

Now if you compare this checksum with the one obtained in the packet you will find that both are exactly same and hence the IP header's integrity was not lost.

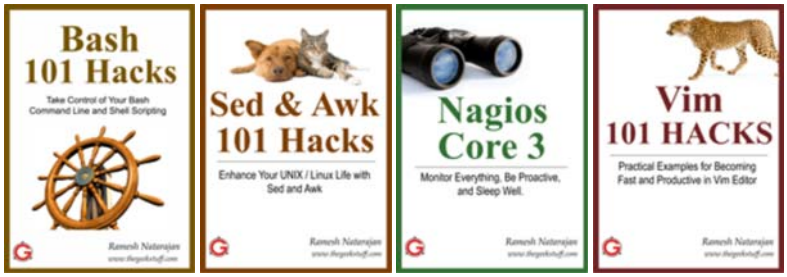
So this is the way we calculate IP header checksum to check the integrity of IP header.

> Add your comment

If you enjoyed this article, you might also like..

1. 50 Linux Sysadmin Tutorials
2. 50 Most Frequently Used Linux Commands (With Examples)
3. Top 25 Best Linux Performance Monitoring and Debugging Tools
4. Mommy, I found it! – 15 Practical Linux Find Command Examples
5. Linux 101 Hacks 2nd Edition eBook Free

- Awk Introduction – 7 Awk Print Examples
- Advanced Sed Substitution Examples
- 8 Essential Vim Editor Navigation Fundamentals
- 25 Most Frequently Used Linux IPTables Rules Examples
- Turbocharge PuTTY with 12 Powerful Add-Ons



{ 26 comments… add one }

- Jalal Hajigholamali May 17, 2012, 2:49 am

  Hi,

  Thanks again for very nice article…

  Link
- bob May 17, 2012, 7:29 am

  Thanks again for a good article. Examples are great.

  Link
- Ravi May 17, 2012, 8:08 am

  Hi,
  Thanks again for very nice article…

  Link
- Pierre B. May 17, 2012, 9:28 am

  TGS strikes again!

  This IS a good tutorial, definitely a good one! Thanks for your effort. In case some of your readers want some more information about Checksums, encryption etc. be sure to check this one out . It is also a good one.

  Thx Arora.

  Link
- Ran May 17, 2012, 2:23 pm

  RGI – Really Good Information!
  Thanks for the time,

  Link
- rajesh May 17, 2012, 9:53 pm

  Great article. cheers