Nmap

Forensic Investigation of Nmap Scan using Wireshark

January 17, 2018 By Raj Chandel

Today we are discussing how to read hexadecimal bytes from an IP Packet that helps a network admin to identify various types of NMAP scanning. But before moving ahead please read our previous both articles "**Network packet forensic**" and "**NMAP scanning with Wireshark**" it will help you in a better understanding of this article.

Requirement

Attacking Tool: Nmap

Analysis Tool: Wireshark

We are going to calculate hexadecimal bytes of Wireshark using given below table and as we know Wireshark capture network packet mainly of 4 layers which are described below in table as per OSI layer model and TCP/IP layer model.

Layer Captured by Wireshark	TCP/IP layer as per Wireshark	OSI layer as per Wireshark
Ethernet Header	L1 Network Interface Layer	L2 Data Link Layer
IP Header	L2 Internet Layer	L3 Network Layer
TCP/UDP Header	L3 Transport Layer	L4 Transport layer
Application Header	L4 Application Layer	L7 Applcation Layer

Nmap ARP Scanning

Let 's start!!

Hopefully, the reader must be aware of basic NMAP scanning techniques if not then read it from here, now open the terminal and execute given below command which known as "HOST SCAN" to identify a live host in the network.

```
nmap -sn 192.168.1.100
```

Nmap uses the –sP/-sn flag for host scans and broadcasts ARP request packet to identify which IP is allocated to the particular host machine. From given below image you can observe that "1 host up" message.

Working of ARP Scan for Live Host

1. Send ARP request for MAC address
2. Receive MAC address through ARP Reply packet

```
root@kali:~# nmap -sn 192.168.1.100

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-09 03:00 EST
Nmap scan report for 192.168.1.100
Host is up (0.00016s latency).
MAC Address: FC:AA:14:6A:9A:A2 (Giga-byte Technology)
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
root@kali:~#
```

Step to Identify Nmap ARP Scan

- **Collect Ethernet Header details**

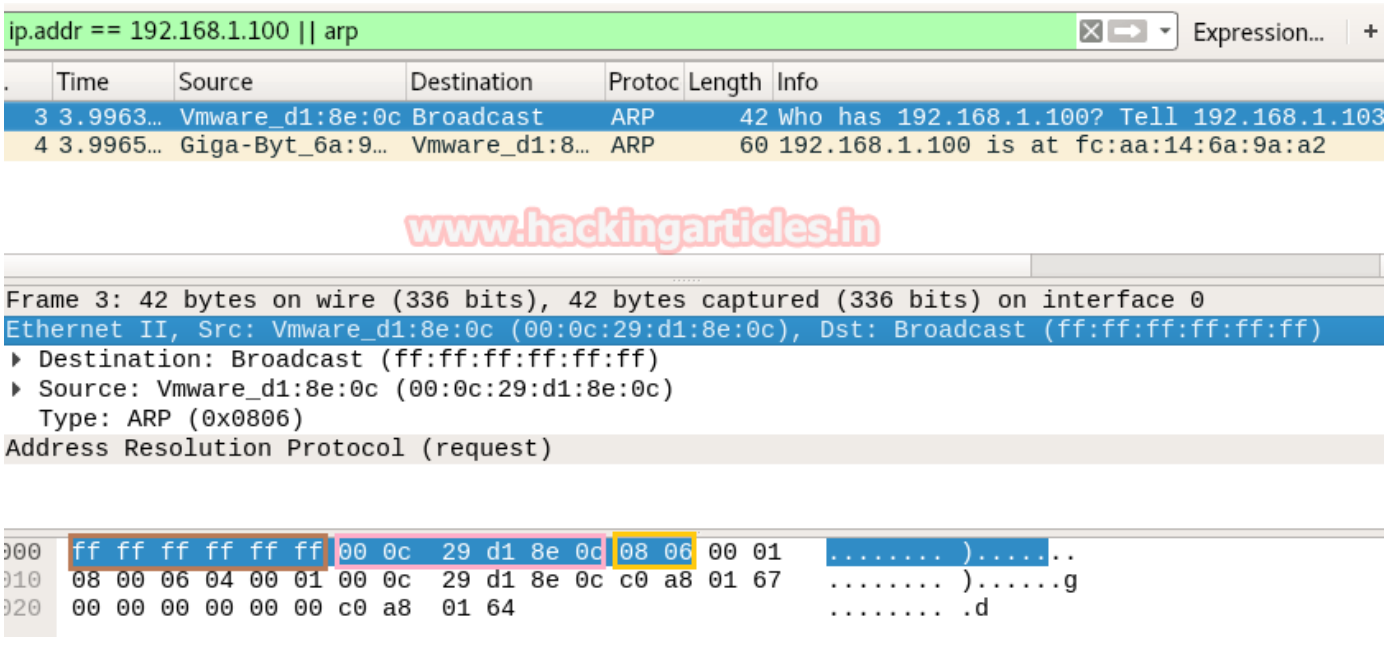
Here we used Wireshark to capture the network packet coming from victim's network

order to analysis only ARP packet we have applied filter "**ip.addr == VICTIM IP || arp**" as shown in given below image. Here you will find 2 arp packets, basically, the 1st arp packet is broadcasting IP for asking MAC address of that network and the 2nd packet is unicast contains Answer of IP query.

Now let's read Hex value of Ethernet header for identifying source and destination Mac addresses along with that we can also enumerate the bytes used for an encapsulated packet, in order to identify Ether type is being used here.

Ethernet header 14 bytes	Destination MAC Address 6 Bytes	Source MAC Address 6 Bytes	Ether Type 2 Bytes
Bits Color	**Brown**	**Pink**	**Yellow**
Hexadecimal value	ff:ff:ff:ff:ff:ff	00:0c:29:d1:8e:0c	0806

Hence from Ethernet header, we can conclude it as ARP broadcast packet asking for destination Mac address. There shouldn't be any uncertainty in concern with source Mac address who is responsible for sending packet but if we talk about Destination Mac address then we got ff:ff:ff:ff:ff:ff:ff which means exact Destination is the machine is not available here. Further moving ahead we found **Ether type 0x0806** highlighted in yellow colour is used for ARP protocol.



Collect ARP Header (Request/Reply)

In order to identify ARP scan, you need to investigate some important parameters which could help a network admin to make a correct assumption in concern of ARP scan.

Try to collect the following details as given below:

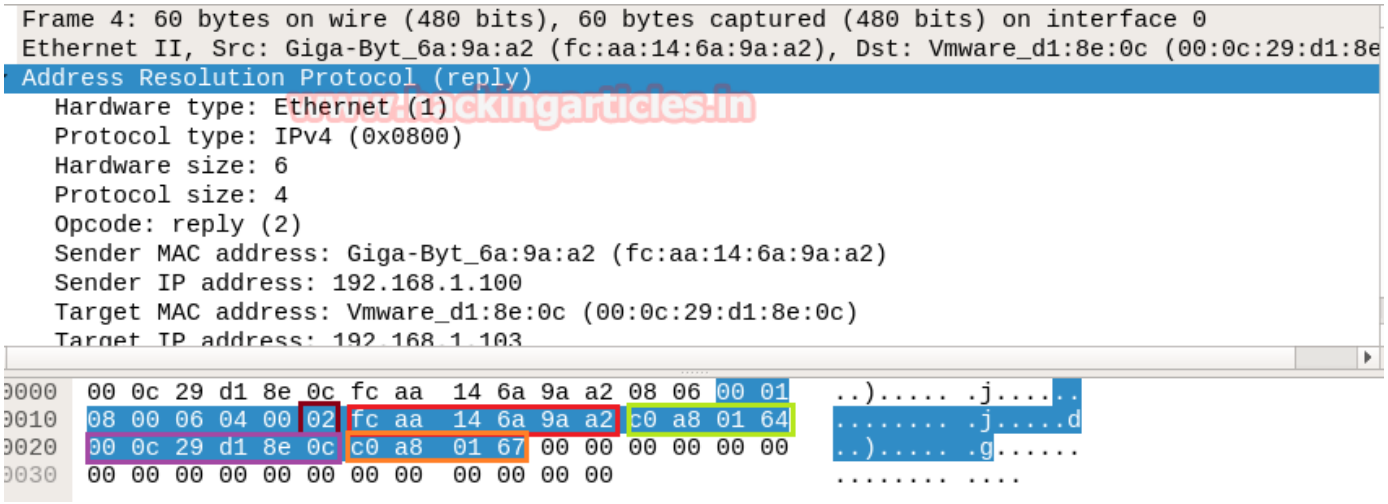
- Opcode (Request/Reply)
- Source Mac
- Source IP
- Destination MAC
- Destination IP

```
Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: Vmware_d1:8e:0c (00:0c:29:d1:8e:0c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: request (1)
   Sender MAC address: Vmware_d1:8e:0c (00:0c:29:d1:8e:0c)
   Sender IP address: 192.168.1.103
   Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
   Target IP address: 192.168.1.100
```

```
000  ff ff ff ff ff ff 00 0c  29 d1 8e 0c 08 06 00 01    ........ ).......
010  08 00 06 04 00 01 00 0c  29 d1 8e 0c c0 a8 01 67    ........ ).......g
020  00 00 00 00 00 00 c0 a8  01 64                      ........ .d
```

Now with help of the following table, you can read the hex value highlighted in above and below image for ARP Request and Reply packets respectively.

ARP Header =>	Opcode	Source Mac	Source IP	Destination MAC	Destination IP
Bits Color	Brown	Red	Green	Purple	Orange
ARP Request Hex Value	01	00:0c:29:d1:8e:0c	C0.a8.01.67	00:00:00:00:00:00	C0.a8.01.64
Decimal value of Request	1	No need	192.168.1.103	No need	192.168.1.100
ARP Reply Hex Value	02	Fc:aa:14:6a:9a:a2	C0.a8.01.64	00:0c:29:d1:8e:0c	C0.a8.01.67
Decimal Value of Reply	2	No need	192.168.1.100	No need	192.168.1.103



Nmap ICMP Scanning

Now execute given below command which known as “HOST SCAN” to identify a live host in a network by sending **Ping request** with the help of ICMP packet.

```
nmap -sn 192.168.1.100 –disable-arp-ping
```

Now above command will send ICMP request packet instead of ARP request for identifying the live host in a network.

Working of NMAP ICMP Ping when a host is live:

1. Send ICMP echo **request** packet.
2. Receive ICMP echo **reply.**

- Send **TCP SYN** packet on any TCP port (this port must be rarely blocked by network admin).

1. Receive **TCP RST-ACK** from target’s Network.

As a result, NMAP gives “HOST UP” message as shown in given below image.

```
root@kali:~# nmap -sn 192.168.1.100 --disable-arp-ping

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-09 04:58 EST
Nmap scan report for 192.168.1.100
Host is up (0.00018s latency).
MAC Address: FC:AA:14:6A:9A:A2 (Giga-byte Technology)
Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
```

Step to Identify NMAP ICMP Scan

- **Collect IP Header Details for Protocol version**

For reading data of Ethernet head visit to our previous article “Network packet forensic”.

NOTE: Ether type for IPv4 is 0x0800

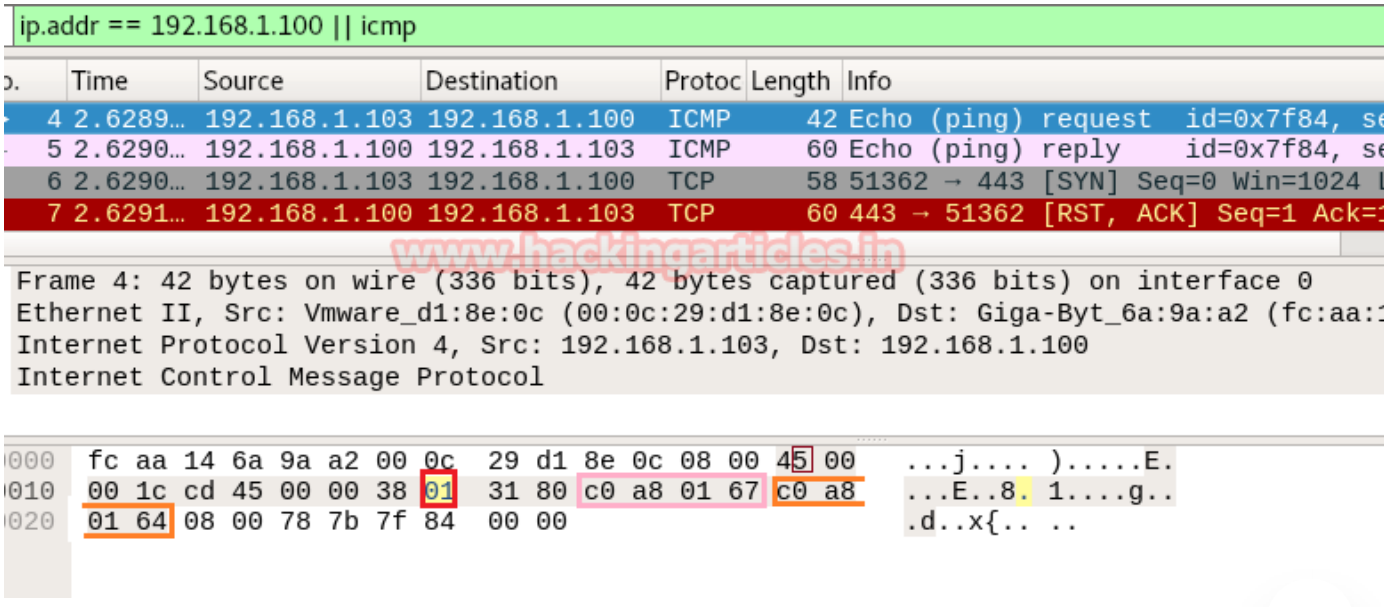
Since we know ICMP is Layer 3 protocol according to the OSI model, therefore, we need to focus on following details for ICMP forensic with help of IP Header of a packet.

Try to collect the following details as given below:

1. Ip header length 20 Bytes (5bits*4=20 bytes)
2. Protocol (01 for ICMP)
3. Source IP
4. Destination IP

From given below image you can observe Hexadecimal information of IP header field and using the given table you can study these value to obtain their original value.

IP header (20 bytes)	Header length	Protocol	Source IP	Destination IP
Bits Color	**Brown**	**Red**	**Pink**	**Orange**
Hex Value	5	01	C0.a8.01.67	C0.a8.01.64
Decimal value	5	1	192.168.1.103	192.168.1.100



The IP header length is always given in form of the bit and here it is 5 bit which is also

minimum IP header length and to make it 20 bytes multiple 5 with 4 i.e. 5*4 bytes =20 bytes.

Identify ICMP Message type (Request /Reply)

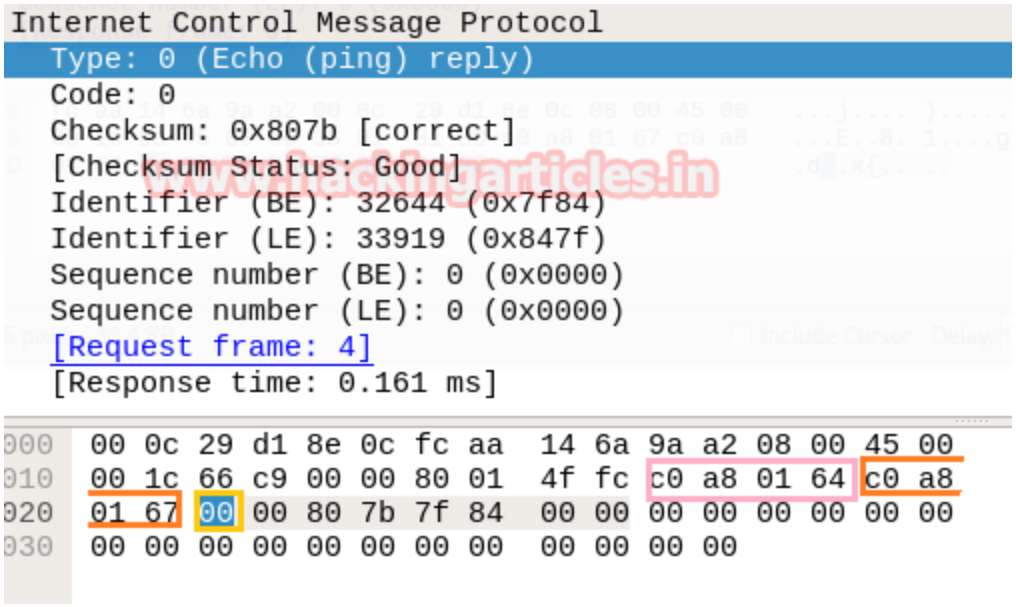
Now we had discussed above according to Nmap ICMP scanning technique the **1st packet** is should be **ICMP echo request** packet and **a 2nd packet** is should be of **ICMP echo reply** packet.

```
Internet Control Message Protocol
   Type: 8 (Echo (ping) request)
   Code: 0
   Checksum: 0x787b [correct]
   [Checksum Status: Good]
   Identifier (BE): 32644 (0x7f84)
   Identifier (LE): 33919 (0x847f)
   Sequence number (BE): 0 (0x0000)
   Sequence number (LE): 0 (0x0000)
   [Response frame: 5]
```

```
000   fc aa 14 6a 9a a2 00 0c  29 d1 8e 0c 08 00 45 00
010   00 1c cd 45 00 00 38 01  31 80 c0 a8 01 67 c0 a8
020   01 64 08 00 78 7b 7f 84  00 00
```

Now with help of the following table, you can read hex value highlighted in above and below image for ICMP Request and Reply packets respectively.

IP Header =>	ICMP Type	Source IP	Destination IP
Bits color	Yellow	Pink	Orange
ICMP Echo Request Hex Value	08	C0.a8.01.67	C0.a8.01.64
Decimal value of Request	8	192.168.1.103	192.168.1.100
ICMP Echo Reply Hex Value	00	C0.a8.01.64	C0.a8.01.67
Decimal Value of Reply	0	192.168.1.100	192.168.1.103



Identify TCP Flags

AS discussed above after ICMP reply, the **3rd packet** should be **of TCP-SYN** packet and **4th** should be of **TCP-RST/ACK**. We had seen in our previous article the hex value of all TCP-Flags are different from each other, so if we are talking for TCP-SYN flag then its Hex value should 0x02.

From given below table you can observe the sequence of TCP flag and how bits of these flags are set for sending the packet to the destination port.

For example, if you found TCP SYN packet then the bit for **SYN flag** is set **1** for which the binary value will be **000000010** and its hexadecimal will be **0x02.**

NS	CWR	ECE	URG	ACK	PSH	RST	**SYN**	FIN
0	0	0	0	0	0	0	**1**	0

Sometime you will get the combination of two or more flag in TCP header, so in that scenario take the help of the following table to read the Hex value of such packet to identify TCP flags bits are being set 1.

For example, if you found **TCP SYN/ACK** packets then indicates that SYN & ACK flags are set 1 for which the binary value will be **000010010** and its hexadecimal will be **0x12**

NS	CWR	ECE	URG	**ACK**	PSH	RST	**SYN**	FIN
0	0	0	0	**1**	0	0	**1**	0

Therefore I design below table to let you know more about of Hex value when two or more

than two flags are set 1.

TCP Flag	Decimal Value	HexValue
SYN + ACK	2 + 16 = 18	2 + 10 = 12
RST + ACK	4 + 16 = 20	4 + 10 = 14
PSH + ACK	8 + 16 = 24	8 + 10 = 18
FIN + PSH + URG	1 + 8 + 32 = 41	1+ 8 + 20 = 29
URG	32	20
ACK	16	10
PSH	8	08
RST	4	04
SYN	2	02
FIN	1	01


```
Frame 6: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface
Ethernet II, Src: Vmware_d1:8e:0c (00:0c:29:d1:8e:0c), Dst: Giga-Byt_6a:9a:a2 (
Internet Protocol Version 4, Src: 192.168.1.103, Dst: 192.168.1.100
Transmission Control Protocol, Src Port: 51362, Dst Port: 443, Seq: 0, Len: 0
```



```
000  fc aa 14 6a 9a a2 00 0c  29 d1 8e 0c 08 00 45 00   ...j.... )....E.
010  00 2c fa 3e 00 00 33 06  09 72 c0 a8 01 67 c0 a8   .,.>..3. .r...g..
020  01 64 c8 a2 01 bb bc af  75 68 00 00 00 00 60 02   .d...... uh....`.
030  04 00 13 95 00 00 02 04  05 b4                     ........ ..
```

The image given above contains the hex value of **TCP-SYN** packets and the image given below contains the hex value of **TCP-RST/ACK** packet from which we can calculate the source port and the destination port of the packet respectively like one given below.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP –SYN Packets Hex value	C8 a2	01 bb	02
Decimal Value	51362	443	2
TCP-RST/ACK packet Hex value	01 bb	C8 a2	14
Decimal Value	443	51362	20

Conclusion! So as stated above regarding the working of NMAP ICMP scan, we had obtained the hex value for every packet in the same sequence. Obtaining the hex value for every packet in such sequence gives the indication to the Penetration tester that Someone has Choose NMAP ICMP scan for Network enumeration.

```
Transmission Control Protocol, Src Port: 443, Dst Port: 51362, Seq: 1, Ack: 1,
```



```
000  00 0c 29 d1 8e 0c fc aa  14 6a 9a a2 08 00 45 00   ..)..... .j....E.
010  00 28 66 ca 40 00 80 06  0f ea c0 a8 01 64 c0 a8   .(f.@... .....d..
020  01 67 01 bb c8 a2 00 00  00 00 bc af 75 69 50 14   .g...... ....uiP.
030  00 00 2f 3e 00 00 00 00  00 00 00 00               ../>.... ....
```

Default NMAP Scan (Stealth Scan)

Here we are going with the default scan method to enumerate the “open” state of any specific port

```
nmap -p80 192.168.1.100
```

Working of Default Scan for open port:

1. Send TCP-SYN packet
2. Receive TCP-SYN/ACK
3. Send TCP-RST packet

It is also known as half Open TCP Scan as it does not send ACK packet after receive SYN/ACK packet.

```
root@kali:~# nmap -p80 192.168.1.100

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-09 09:06 EST
Nmap scan report for 192.168.1.100
Host is up (0.00018s latency).

PORT   STATE SERVICE
80/tcp open  http
MAC Address: FC:AA:14:6A:9A:A2 (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

Step to Identify NMAP Default Scan (Stealth Scan)

- **Collect IP Header Details for Protocol Version**

For reading data of Ethernet head visit to our previous article “Network packet forensic

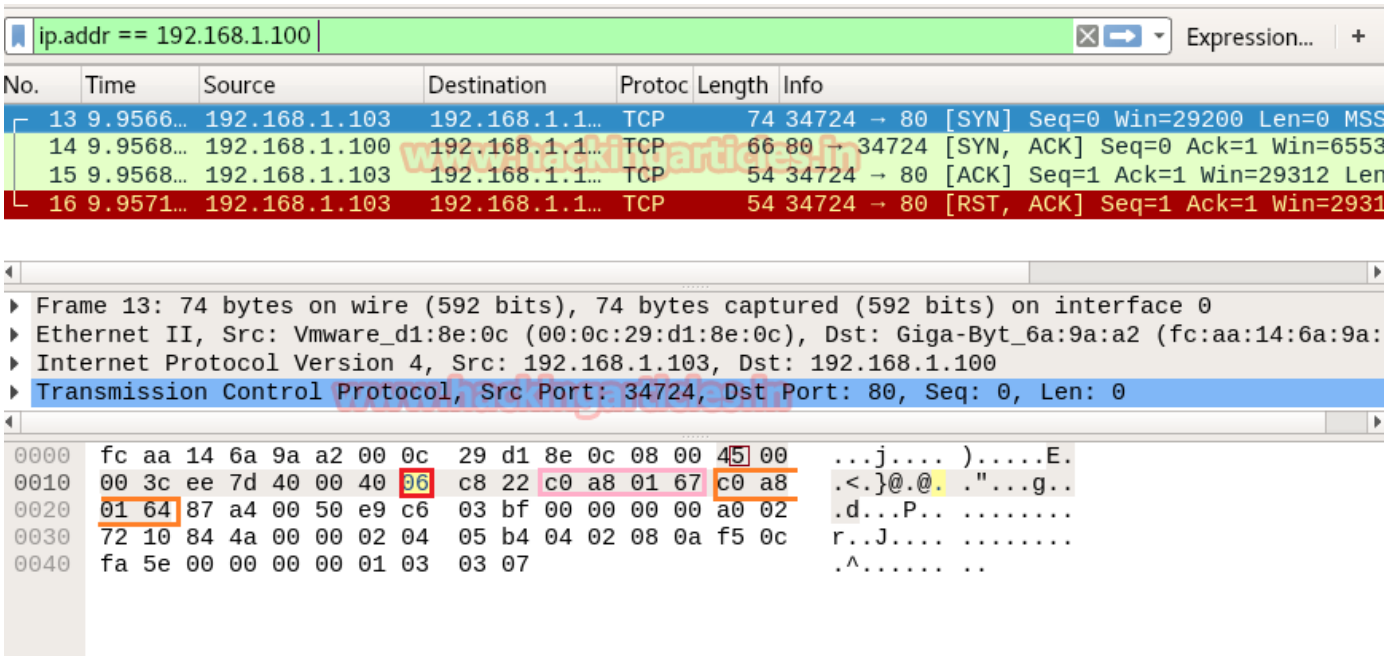
NOTE: Ether type for IPv4 is 0x0800.

Try to collect the following details as given below:

1. Ip header length 20 Bytes (5bits*4=20 bytes)
2. Protocol (6 for TCP)
3. Source IP
4. Destination IP

IP header (20 bytes)	Header length	Protocol	Source IP	Destination IP
Bits Color	**Brown**	**Red**	**Pink**	**Orange**
Hex Value	5	06	C0.a8.01.67	C0.a8.01.64
Decimal value	5	6	192.168.1.103	192.168.1.100

From given below image you can observe Hexadecimal information of the IP header field and using the given table you can study these value to obtain their original value.



Analysis TCP Header Details

Since from the above image we had to obtain Source and Destination IP and protocol used for communication i.e. TCP, now we need to identify the source and Destination port and TCP Flag used for establishing the connection between two systems.

In the image we have highlighted source port in “Light brown” colour and destination port in “yellow colour”, you can use given below table to read the hex value of the given image.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP–SYN Packets Hex value	92 62	00 50	0x02
Decimal Value	38498	80	2

So we come to know that here **TCP-SYN** packet is used for sending connection request on Port 80.

```
Transmission Control Protocol, Src Port: 38498, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 38498
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0    (relative sequence number)
  Acknowledgment number: 0
  0110 .... = Header Length: 24 bytes (6)
▸ Flags: 0x002 (SYN)
  Window size value: 1024
  [Calculated window size: 1024]
  Checksum: 0x01f6 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
▸ Options: (4 bytes), Maximum segment size
```

```
0000  fc aa 14 6a 9a a2 00 0c  29 d1 8e 0c 08 00 45 00   ...j.... ).....E.
0010  00 2c ea 8e 00 00 38 06  14 22 c0 a8 01 67 c0 a8   .,....8. ."...g..
0020  01 64 96 62 00 50 56 0b  21 57 00 00 00 00 60 02   .d.b.PV. !W....`.
0030  04 00 01 f6 00 00 02 04  05 b4                     ........ ..
```

Again we read next packet then here we found **hex value 12** indicates that **TCP-SYN/ACK** has been sending from port 80.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP–SYN/ACK Packets Hex value	00 50	92 62	0x12
Decimal Value	80	38498	18

Take the help given above table to read the hex value of the given image. Hex value 12 for TCP flag is used for SYN + ACK as explained above, and we get **0x12** by adding Hex value “ 0x02 of SYN” and “0x10 of ACK”.

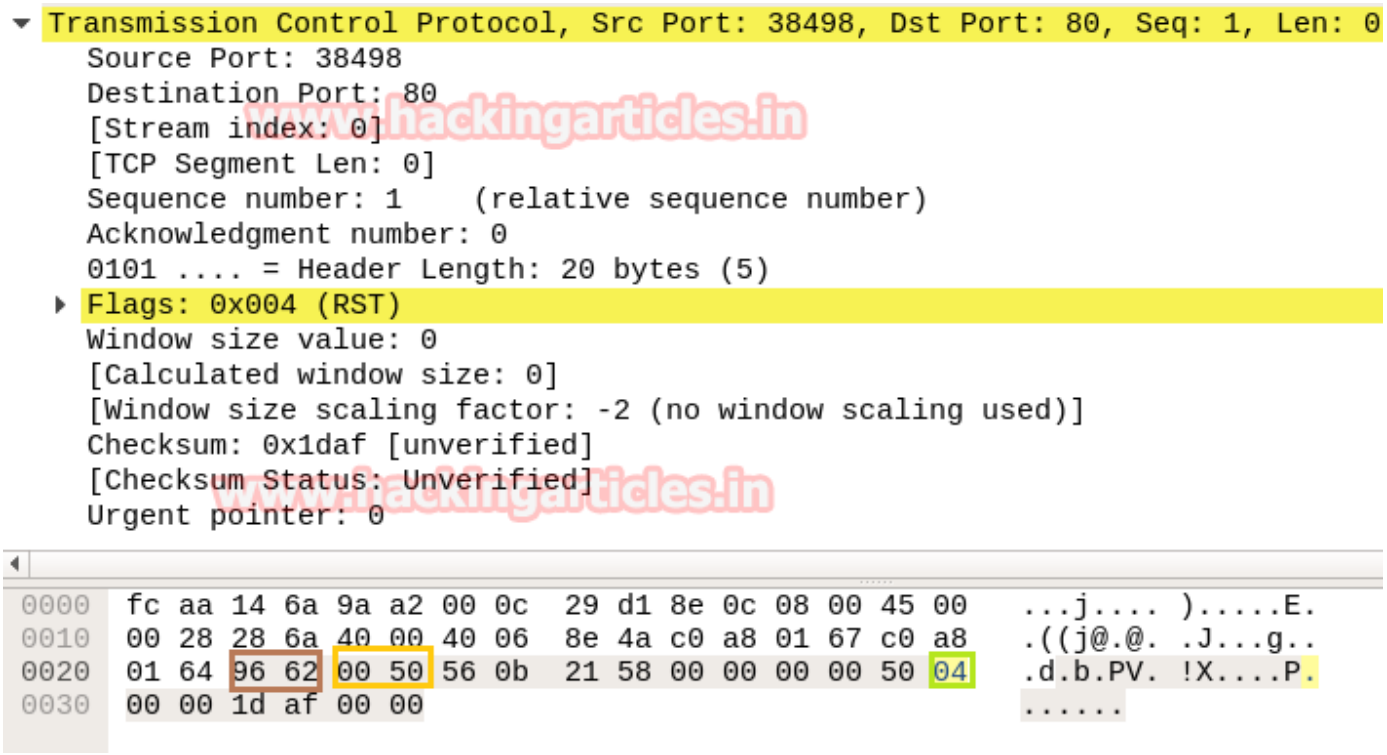
```
Transmission Control Protocol, Src Port: 80, Dst Port: 38498, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 38498
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0    (relative sequence number)
  Acknowledgment number: 1    (relative ack number)
  0110 .... = Header Length: 24 bytes (6)
▸ Flags: 0x012 (SYN, ACK)
  Window size value: 64240
  [Calculated window size: 64240]
  Checksum: 0x11c5 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
▸ Options: (4 bytes), Maximum segment size
▸ [SEQ/ACK analysis]
```

```
0000  00 0c 29 d1 8e 0c fc aa  14 6a 9a a2 08 00 45 00   ..)..... .j....E.
0010  00 2c 69 27 40 00 80 06  0d 89 c0 a8 01 64 c0 a8   .,i'@... .....d..
0020  01 67 00 50 96 62 17 52  e1 dc 56 0b 21 58 60 12   .g.P.b.R ..V.!X`.
0030  fa f0 11 c5 00 00 02 04  05 b4 00 00               ........ ....
```

In the image given below, we come to know that **TCP-RST** packet is used for sending Reset connection to Port 80.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP –RST Packets Hex value	96 62	00 50	0x04
Decimal Value	38498	80	4

Conclusion! So as declared above regarding the working of NMAP default scan or NMAP stealth scan we had to obtain the hex value for every packet in the same sequence. Obtaining the hex value for every packet in such sequence gives an indication to the Penetration tester that Someone has Choose NMAP Default scan for Network enumeration.



Nmap TCP Scan

Here we are going with TCP scan to enumerate state of any specific port

```
nmap -sT -p80 192.168.1.100
```

Working of Default Scan for open port:

1. Send TCP-SYN packet
2. Receive TCP-SYN/ACK

1. Send TCP-ACK packet
2. Send TCP-RST/ACK packet

```
root@kali:~# nmap -sT -p80 192.168.1.100

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-09 03:09 EST
Nmap scan report for 192.168.1.100
Host is up (0.00018s latency).
PORT   STATE SERVICE
80/tcp open  http
MAC Address: FC:AA:14:6A:9A:A2 (Giga-byte Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```


Step to Identify NMAP TCP Scan

- **Collect IP Header Details for Protocol Version**

For reading data of Ethernet head visit to our previous article “Network packet forensic”.

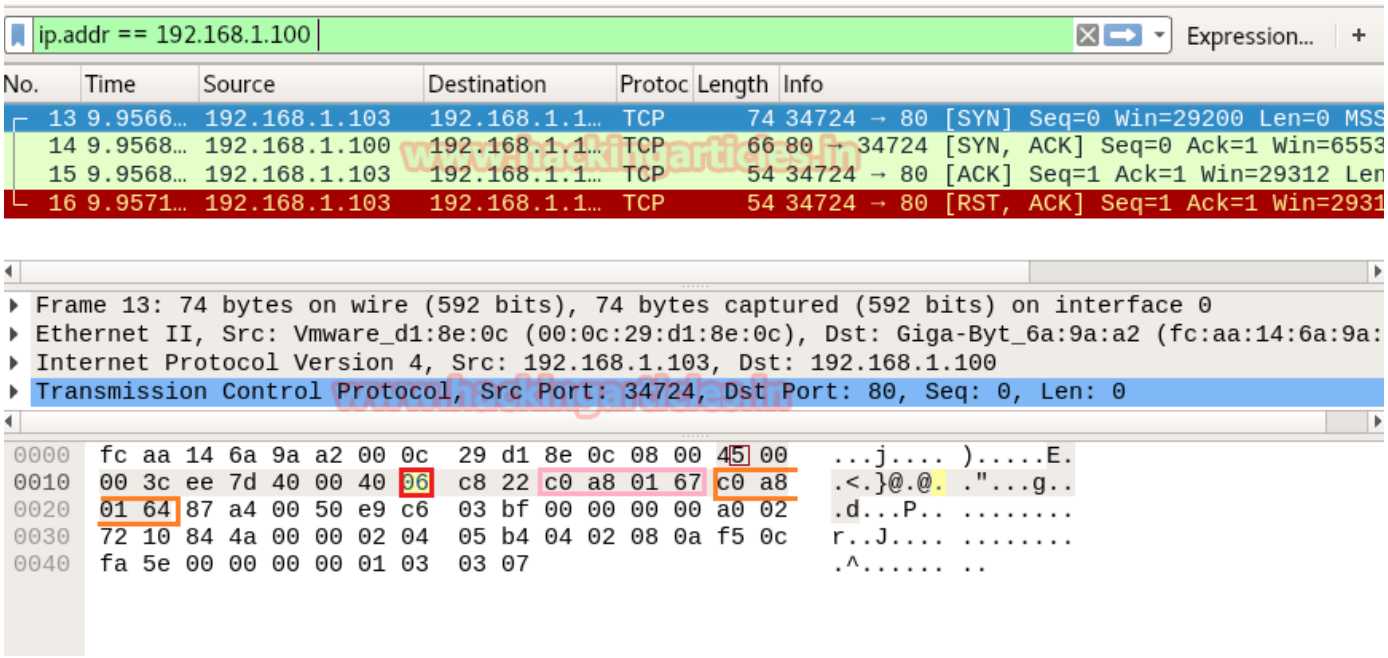
NOTE: Ether type for IPv4 is 0x0800.

Try to collect the following details as given below:

1. Ip header length 20 bytes (5bits*4=20 bytes)
2. Protocol (06 for TCP)
3. Source IP
4. Destination IP

IP header (20 bytes)	Header length	Protocol	Source IP	Destination IP
Bits Color	**Brown**	**Red**	**Pink**	**Orange**
Hex Value	5	06	C0.a8.01.67	C0.a8.01.64
Decimal value	5	6	192.168.1.103	192.168.1.100

It is quite similar to NMAP stealth Scan and using a given table you can study these values to obtain their original value.



- **Analysis TCP Header Details**

NMAP TCP Scan follows **3-way handshake of TCP** connection for enumeration open port. Identifying source and destination port along with Flag hex value (**TCP-SYN**) are similar as above.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP –SYN Packets Hex value	87 a4	00 50	0x02
Decimal Value	34724	80	2

So we come to know that here **TCP-SYN** packet is used for sending connection request on Port 80.

```
Transmission Control Protocol, Src Port: 34724, Dst Port: 80, Seq: 0, Len: 0
   Source Port: 34724
   Destination Port: 80
   [Stream index: 0]
   [TCP Segment Len: 0]
   Sequence number: 0    (relative sequence number)
   Acknowledgment number: 0
   1010 .... = Header Length: 40 bytes (10)
 ▸ Flags: 0x002 (SYN)
   Window size value: 29200
   [Calculated window size: 29200]
   Checksum: 0x844a [unverified]
   [Checksum Status: Unverified]
   Urgent pointer: 0
 ▸ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation
```

```
0000  fc aa 14 6a 9a a2 00 0c  29 d1 8e 0c 08 00 45 00   ...j.... ).....E.
0010  00 3c ee 7d 40 00 40 06  c8 22 c0 a8 01 67 c0 a8   .<.}@.@. ."...g..
0020  01 64 87 a4 00 50 e9 c6  03 bf 00 00 00 00 a0 02   .d...P.. ........
0030  72 10 84 4a 00 00 02 04  05 b4 04 02 08 0a f5 0c   r..J.... ........
0040  fa 5e 00 00 00 00 01 03  03 07                     .^...... ..
```

Again we read next packet then here we found hex value 12 indicates that **TCP-SYN/ACK** has been sent via port 80.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP –SYN/ACK Packets Hex value	00 50	87 a4	12
Decimal Value	80	34724	18

```
Transmission Control Protocol, Src Port: 80, Dst Port: 34724, Seq: 0, Ack: 1, Len:
   Source Port: 80
   Destination Port: 34724
   [Stream index: 0]
   [TCP Segment Len: 0]
   Sequence number: 0    (relative sequence number)
   Acknowledgment number: 1    (relative ack number)
   1000 .... = Header Length: 32 bytes (8)
 ▸ Flags: 0x012 (SYN, ACK)
   Window size value: 65535
   [Calculated window size: 65535]
   Checksum: 0xae76 [unverified]
   [Checksum Status: Unverified]
   Urgent pointer: 0
 ▸ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-
 ▸ [SEQ/ACK analysis]
```

```
0000  00 0c 29 d1 8e 0c fc aa  14 6a 9a a2 08 00 45 00   ..)..... .j....E.
0010  00 34 52 33 40 00 80 06  24 75 c0 a8 01 64 c0 a8   .4R3@... $u...d..
0020  01 67 00 50 87 a4 ec 9c  da 55 e9 c6 03 c0 80 12   .g.P.... .U......
0030  ff ff ae 76 00 00 02 04  05 b4 01 03 03 08 01 01   ...v.... ........
0040  04 02                                             ..
```

The only difference between Stealth Scan and TCP scan is that here a packet of ACK fl

sent by source machine who initiate the TCP communication. Again we read next packet then here we found hex value 0x10 indicates that **TCP- ACK** has been sent via port 80.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP –ACK Packets Hex value	87 a4	00 50	10
Decimal Value	34724	80	16

Conclusion! So as stated above regarding the working of NMAP TCP scan, we had obtained the hex value for every packet in the same sequence. Obtaining the hex value for every packet in such sequence gives an indication to the Penetration tester that Someone has Choose NMAP Default scan for Network enumeration.

NOTE: For packet TCP-RST/ACK the hex value will be " 0x14" send by the attacker machine

```
▾ Transmission Control Protocol, Src Port: 34724, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 34724
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x010 (ACK)
    Window size value: 229
    [Calculated window size: 29312]
    [Window size scaling factor: 128]
    Checksum: 0x8436 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
```



```
0000  fc aa 14 6a 9a a2 00 0c  29 d1 8e 0c 08 00 45 00   ...j.... )..... E.
0010  00 28 ee 7e 40 00 40 06  c8 35 c0 a8 01 67 c0 a8   .(.~@.@. .5...g..
0020  01 64 87 a4 00 50 e9 c6  03 c0 ec 9c da 56 50 10   .d...P.. .....VP.
0030  00 e5 84 36 00 00                                 ...6..
```

Nmap FIN Scan

Here we are going with TCP-FIN scan to enumerate "OPEN" state of a particular port in any Linux based system, therefore, execute given below command.

```
nmap -sF -p22 192.168.1.104
```

Working of FIN Scan for open port: Send 2 packets of TCP-FIN on a specific port

FIN is part TCP flag and NMAP used FIN flag to initiate TCP communication instead of following three-way handshake communication.

```
root@kali:~# nmap -sF -p22 192.168.1.104

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-09 08:37 EST
Nmap scan report for 192.168.1.104
Host is up (0.00025s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:6B:71:A7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Step to Identify NMAP FIN Scan

- **Collect IP Header Details for Protocol Version**

For reading data of Ethernet head visit to our previous article “Network packet forensic”.

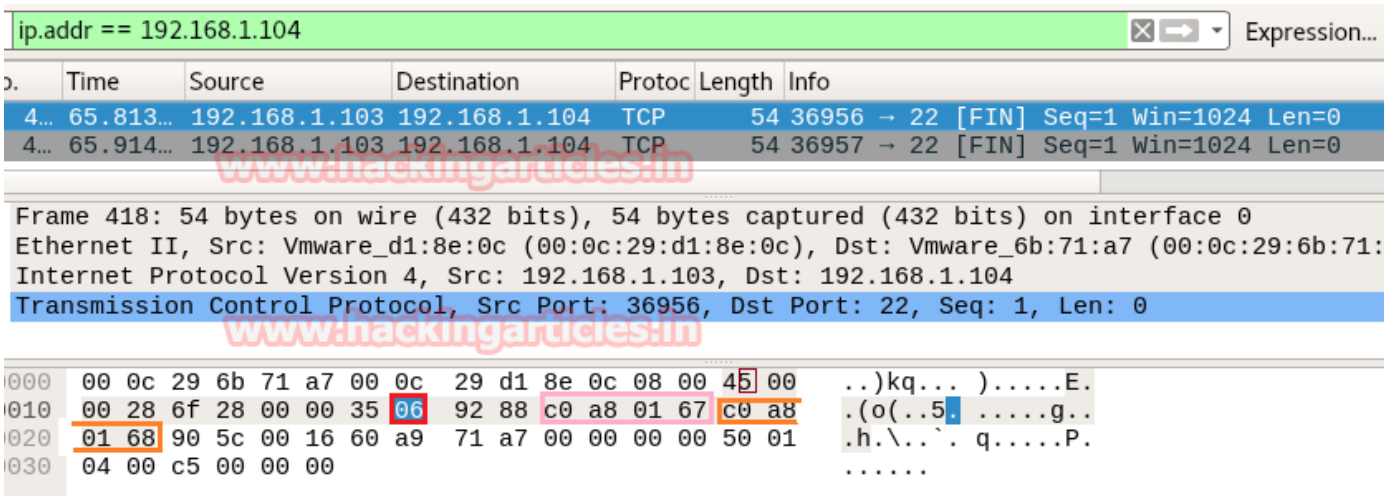
NOTE: Ether type for IPv4 is 0x0800

Try to collect the following details as given below:

1. Ip header length 20 Bytes (5 bits*4=20 bytes)
2. Protocol (06 for TCP)
3. Source IP
4. Destination IP

It is quite similar to NMAP above Scan and using given below table you can study these values to obtain their original value.

IP header (20 bytes)	Header length	Protocol	Source IP	Destination IP
Bits Color	Brown	Red	Pink	Orange
Hex Value	5	06	C0.a8.01.67	C0.a8.01.68
Decimal value	5	6	192.168.1.103	192.168.1.104



- **Analysis TCP Header Details**

Now lets Identifying the source and destination port along with Flag hex value (**TCP-FIN**) is similar as above.

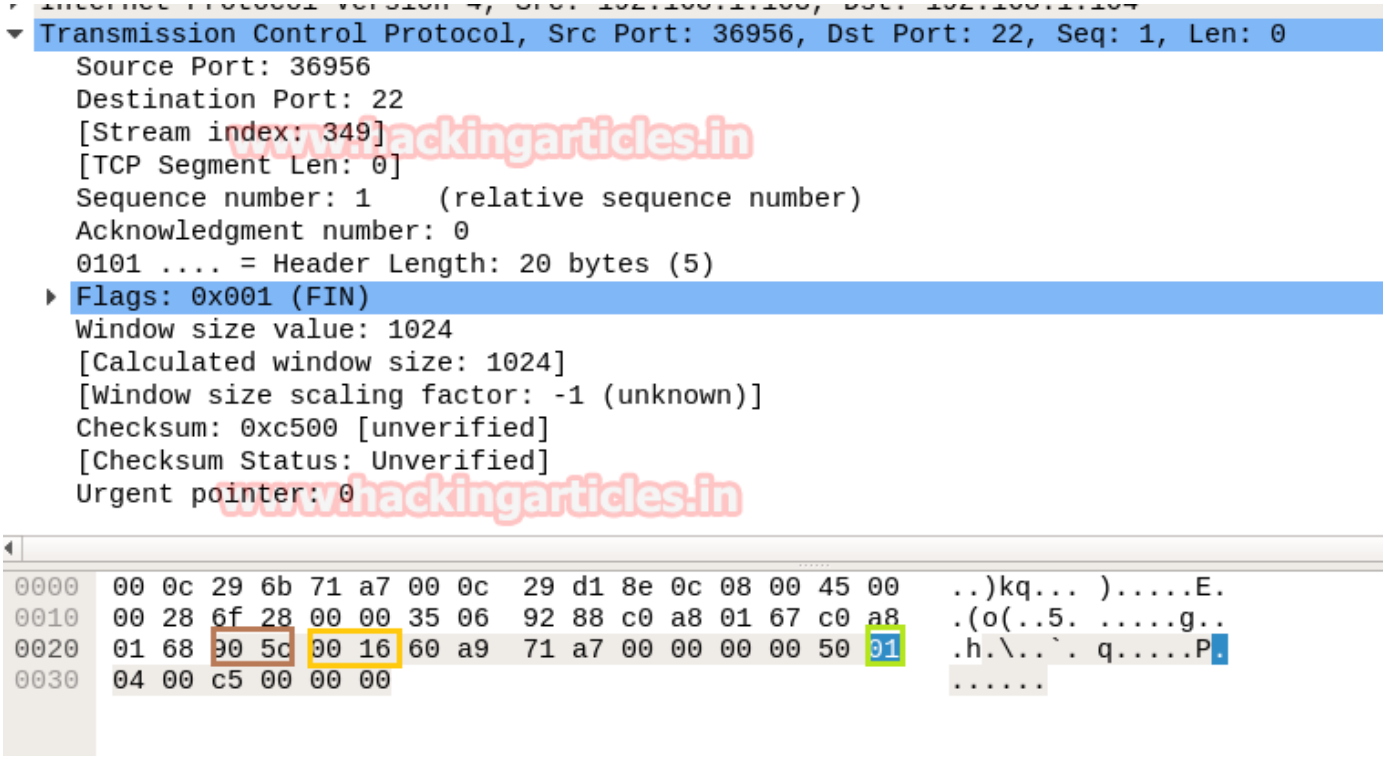
TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP –FIN Packets Hex value	90 5c	00 16	01
Decimal Value	36956	22	1

So through given below image and with help of a table, we came to know that here TCP-FIN packet is used for sending connection request on Port 22.

Conclusion: So as declared above regarding the working of NMAP FIN scan, we had obtained the hex value for every packet in the same sequence.

Obtaining the hex value for every packet in such sequence gives an indication to the Penetration tester that Someone has Choose NMAP FIN scan for Network enumeration.

NOTE: If you found 1st FIN packet (0x01) and 2nd RST packet (0x04) then indicates "Closed Port" on the targeted network.



Nmap NULL Scan

Here we are going with TCP Null scan to enumerate “OPEN” state of any specific port in any Linux based system.

```
nmap -sN -p22 192.168.1.104
```

Working of Null Scan for open port: Send 2 packets of TCP-NONE on a specific port

Here NMAP used NONE flag (No flag) to initiate TCP communication and bit of each flag is set “0” instead of following three-way handshake communication.

Step to Identify NMAP Null Scan

- **Collect IP Header Details for Protocol Version**

For reading data of Ethernet head visit to our previous article “Network packet forensic”.

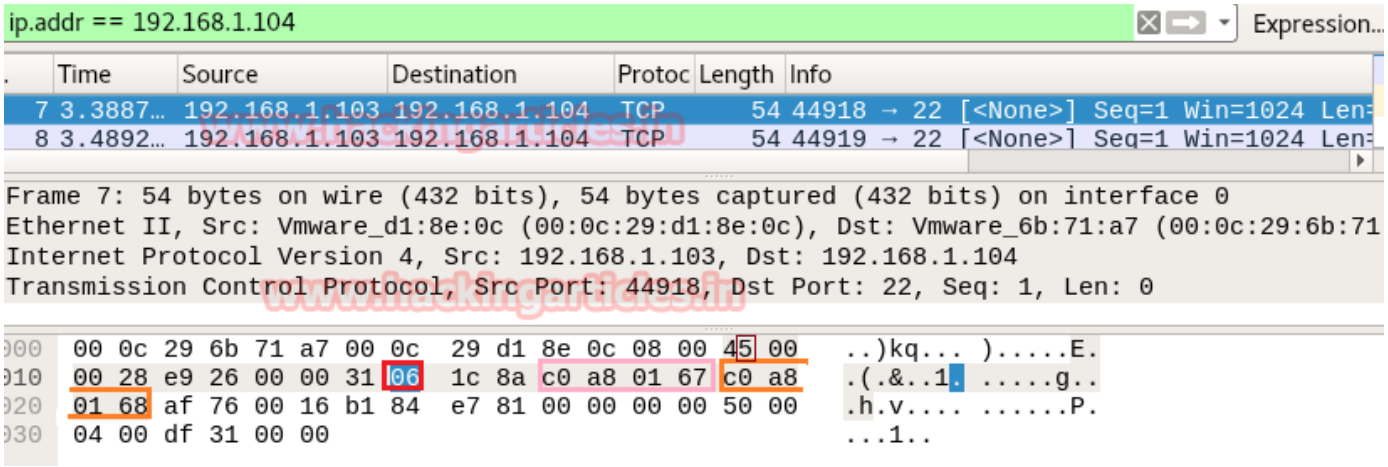
NOTE: Ether type for IPv4 is 0x0800

Try to collect the following details as given below:

1. Ip header length 20 Bytes (5bits*4=20 bytes)
2. Protocol (06 for TCP)
3. Source IP
4. Destination IP

It is quite similar to NMAP above Scan and using the given table you can study these values to obtain their original value.

IP header (20 bytes)	Header length	Protocol	Source IP	Destination IP
Bits Color	**Brown**	**Red**	**Pink**	**Orange**
Hex Value	5	06	C0.a8.01.67	C0.a8.01.68
Decimal value	5	6	192.168.1.103	192.168.1.104



- **Analysis TCP Header Details**

Now lets Identifying the source and destination port along with Flag hex value (**TCP-NONE**) is similar as above.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP-NONE Packets Hex value	Af 76	00 16	0x00
Decimal Value	44918	22	0

So through given below image and with help of a table, we come to know that here TCP-

NONE packet is used for sending connection request on Port 22.

Conclusion: So as stated above regarding the working of NMAP NONE scan, we had obtained the hex value for every packet in the same sequence.

Obtaining the hex value for every packet in such sequence gives an indication to the Penetration tester that someone has Chosen NMAP NONE scan for Network enumeration.

NOTE: If you found 1st NONE packet (0x00) and 2nd RST packet (0x04) then indicates "Closed Port" on the target network.

```
Transmission Control Protocol, Src Port: 44918, Dst Port: 22, Seq: 1, Len: 0
  Source Port: 44918
  Destination Port: 22
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 1    (relative sequence number)
  Acknowledgment number: 0
  0101 .... = Header Length: 20 bytes (5)
▸ Flags: 0x000 (<None>)
  Window size value: 1024
  [Calculated window size: 1024]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0xdf31 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
```

```
000  00 0c 29 6b 71 a7 00 0c  29 d1 8e 0c 08 00 45 00   ..)kq... ).....E.
010  00 28 e9 26 00 00 31 06  1c 8a c0 a8 01 67 c0 a8   .(.&..1. .....g..
020  01 68 af 76 00 16 b1 84  e7 81 00 00 00 00 50 00   .h.v.... ......P.
030  04 00 df 31 00 00                                  ...1..
```

Nmap XMAS Scan

Here we are going with XMAS scan to enumerate "OPEN" state of any specific port in any Linux based system

```
nmap -sX -p22 192.168.1.104
```

Working of XMAS Scan for open port: Send **2 packets of TCP** Flags in a combination of **FIN, PSH, URG** on the specific port.

Here NMAP used 3 TCP flags (FIN, PSH, and URG) to initiate TCP communication and each flag is set "1" instead of following three-way handshake communications.

```
root@kali:~# nmap -sX -p22 192.168.1.104

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-09 08:43 EST
Nmap scan report for 192.168.1.104
Host is up (0.00020s latency).
www.hackingarticles.in
PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:6B:71:A7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Step to Identify NMAP XMAS Scan

- **Collect IP Header Details for Protocol Version**

For reading data of Ethernet head visit to our previous article “Network packet forensic”.

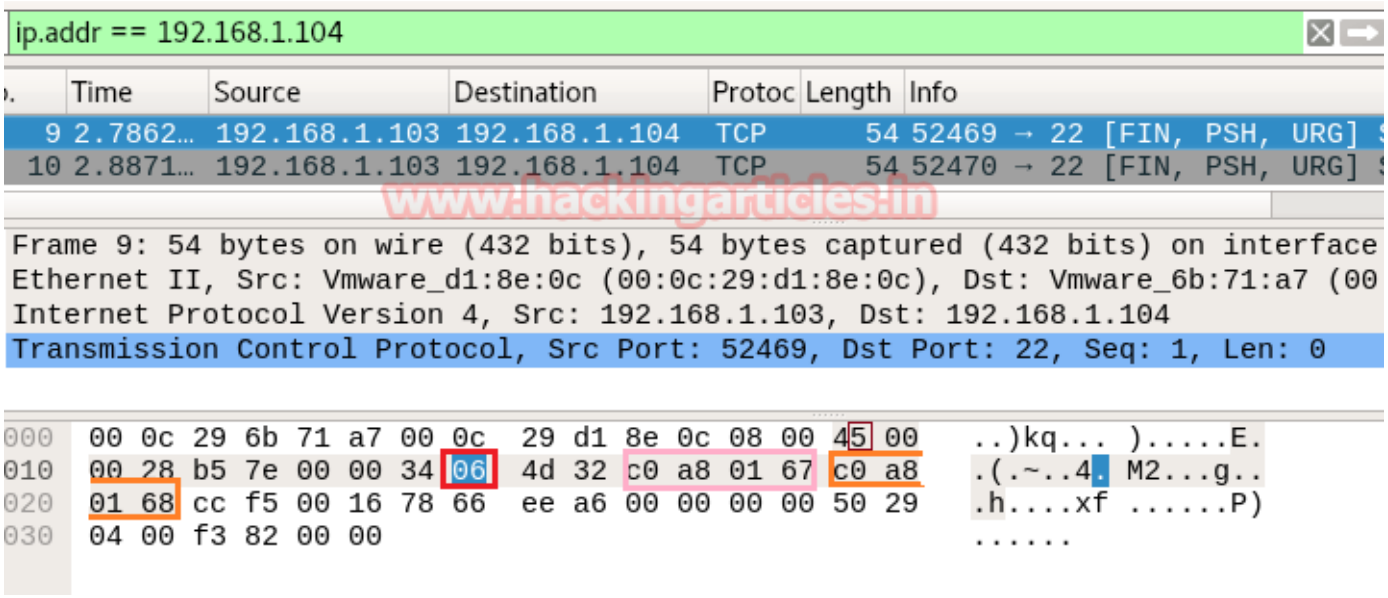
NOTE: Ether type for IPv4 is 0x0800

Try to collect the following details as given below:

1. Ip header length 20 Bytes (5bits*4=20 bytes)
2. Protocol (06 for TCP)
3. Source IP
4. Destination IP

It is quite similar to NMAP above Scan and using the given table you can study these values to obtain their original value.

IP header (20 bytes)	Header length	Protocol	Source IP	Destination IP
Bits Color	**Brown**	**Red**	**Pink**	**Orange**
Hex Value	5	06	C0.a8.01.67	C0.a8.01.68
Decimal value	5	6	192.168.1.103	192.168.1.104



- **Analysis TCP Header Details**

Now lets Identifying the source and destination port along with Flag hex value (**TCP-XMAS**) is similar as above.

TCP Header	Source Port	Destination Port	Hex value of Flag
Bits Color	Light Brown	Yellow	Green
TCP -{FIN,PSH,URG} Packets Hex value	Ccf5	00 16	0x29
Decimal Value	52469	22	41

So through given below image and with help of the table, we come to know that here TCP flags {FIN, PSH, URG} packet is used for sending connection request on Port 22.

Conclusion! So as stated above regarding the working of NMAP XMAS scan, we had obtained the hex value for every packet in the same sequence.

Obtaining the hex value for every packet in such sequence gives the indication to the Penetration tester that someone has Choose NMAP XMAS scanned for Network enumeration.

NOTE:

- If you found 1st {FIN, PSH, URG} packet (0x29) and 2nd RST packet (0x04) then indicate “Closed Port” on targeted network.
- NMAP FIN, NMAP NULL, and NMAP XMAS scan are only applicable on Linux based system

```
Transmission Control Protocol, Src Port: 52469, Dst Port: 22, Seq: 1, Len: 0
   Source Port: 52469
   Destination Port: 22
   [Stream index: 0]
   [TCP Segment Len: 0]
   Sequence number: 1    (relative sequence number)
   Acknowledgment number: 0
   0101 .... = Header Length: 20 bytes (5)
▸  Flags: 0x029 (FIN, PSH, URG)
   Window size value: 1024
   [Calculated window size: 1024]
   [Window size scaling factor: -1 (unknown)]
   Checksum: 0xf382 [unverified]
   [Checksum Status: Unverified]
   Urgent pointer: 0
```

```
0000  00 0c 29 6b 71 a7 00 0c  29 d1 8e 0c 08 00 45 00   ..)kq... ).....E.
0010  00 28 b5 7e 00 00 34 06  4d 32 c0 a8 01 67 c0 a8   .(.~..4. M2...g..
0020  01 68 cc f5 00 16 78 66  ee a6 00 00 00 00 50 29   .h....xf ......P)
0030  04 00 f3 82 00 00                                 ......
```

Nmap UDP Scan

Here we are going with XMAS scan to enumerate state of any specific port in any Linux based system

```
nmap -sU -p68 192.168.1.104
```

Working of XMAS Scan for open port: Send **2 packets of UDP** on a specific port

It is quite different from the TCP communication process because here no Flag is used for establishing a connection or initiate a connection request with the target's network.

```
root@kali:~# nmap -sU -p 68 192.168.1.104

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-09 08:54 EST
Nmap scan report for 192.168.1.104
Host is up (0.00022s latency).

PORT   STATE         SERVICE
68/udp open|filtered dhcpc
MAC Address: 00:0C:29:6B:71:A7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds
```

Step to Identify NMAP UDP Scan

- **Collect IP Header Details for Protocol Version**

For reading data of Ethernet head visit to our previous article “Network packet forensic”.

NOTE: Ether type for IPv4 is 0x0800

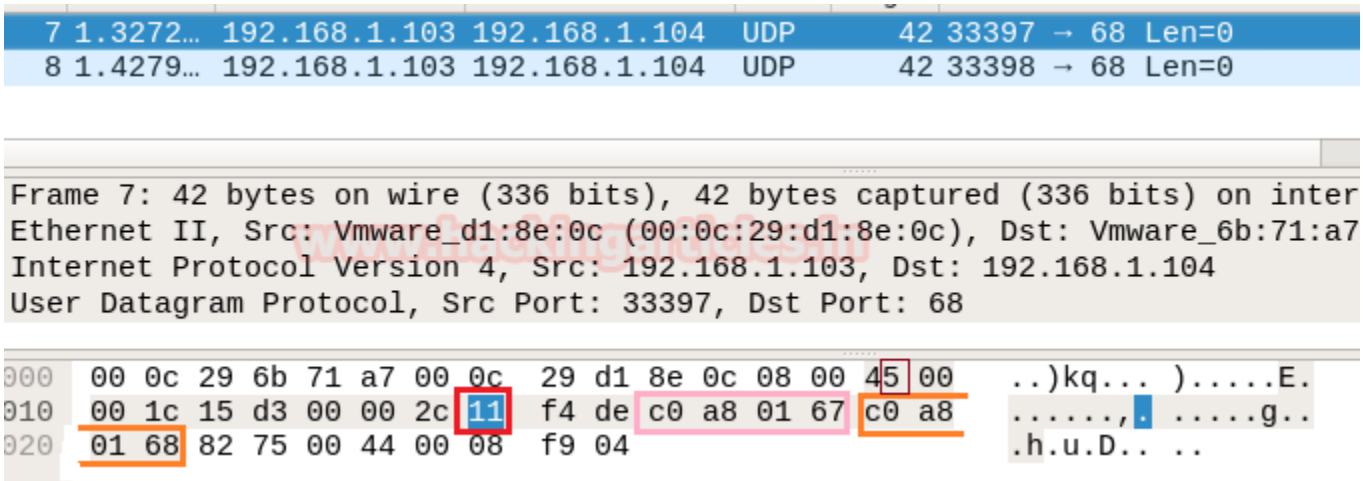
Try to collect the following details as given below:

1. Ip header length 20 Bytes (5 bits*4=20 bytes)
2. Protocol (11 for UDP)
3. Source IP
4. Destination IP

It is quite similar as NMAP above Scan as “IP header” and “Ethernet header” information will be same either is TCP communication or UDP communication and using the given table you can study these values to obtain their original value.

IP header (20 bytes)	Header length	Protocol	Source IP	Destination IP
Bits Color	**Brown**	**Red**	**Pink**	**Orange**
Hex Value	5	11	C0.a8.01.67	C0.a8.01.68
Decimal value	5	17	192.168.1.103	192.168.1.104

Basically, 11 is hex value use for UDP protocol which is quite useful in identify NMAP UDP scan from remanding scanning method.



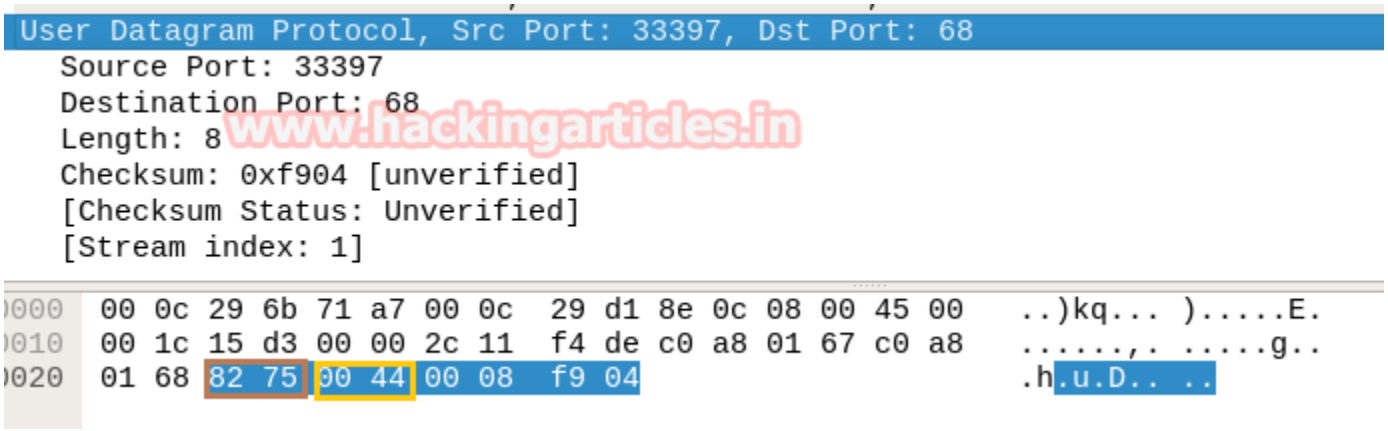
1. **Analysis UDP Header Details**

Now lets Identifying the source and destination port an as done above in TCP Scanning.

TCP Header	Source Port	Destination Port
Bits Color	Light Brown	Yellow
UDP Packets Hex value	82 75	00 44
Decimal Value	3397	68

Conclusion! Obtaining the hex value for every packet in such sequence gives the indication to the Penetration tester that Someone has Choose NMAP UDP scan for Network enumeration.

NOTE: If you found 1st UDP packet and 2nd UDP with ICMP Message Port is unreachable then indicates "Closed Port" on the target network.



Author: **Yashika Dhir** is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact **here**

PREVIOUS POST
Post Exploitation in Windows using dir Command

NEXT POST
Hack the Cyberry: 1 VM(Boot2Root Challenge)

3 thoughts on "Forensic Investigation of Nmap Scan using Wireshark"

Srinivas
August 8, 2019 at 10:14 am

Nice Article

David Mata

October 28, 2019 at 8:05 pm

I am a developer and I always try to use Wireshark to solve problems (related with networking) and with your articles I am going to be able to solve problems that I couln´t before. This articles are just great.

Peter

March 6, 2020 at 3:50 pm

Hi Raj,

Love the article, really great summary and explanation. I often come back to read it.

I just have one question: In 3rd picture in section “Default NMAP Scan (Stealth Scan)” it shows the same Wireshark output as in the 3rd picture in section “Nmap TCP Scan”. In the section “Default NMAP Scan (Stealth Scan)” the Wireshark screenshot should not include an ACK package (it should only be 3 packages in total, not 4 as in the “Nmap TCP Scan”).
Not sure if I expressed it clearly, but I hope you know what I mean.

Cheers,
Peter

Comments are closed.

Search ... Search

Tweets from @hackinarticles

Hacking Articles
@hackinarticles · 1h

Pic of the Day

#infosec #cybersecurity #cybersecuritytips #pentesting #oscp #redteam #informationsecurity #cissp #cybersecuritytips

1 68

Hacking Articles
@hackinarticles · 1h

Pic of the Day

#infosec #cybersecurity #cybersecuritytips #pentesting #oscp #redteam #informationsecurity #cissp #cybersecuritytips

Join Our Training Program

Categories

Cryptography & Stegnography

CTF Challenges

Cyber Forensics

Database Hacking

Footprinting

Hacking Tools

Kali Linux

Nmap

Others

Password Cracking

Penetration Testing

Pentest Lab Setup

Privilege Escalation

Red Teaming

Social Engineering Toolkit

Uncategorized

Website Hacking

Window Password Hacking

Wireless Hacking

Wireless Penetration Testing

Archives

Select Month

You may like