iGNITE Technologies

Forensic Investigation of Nmap Scan with Wireshark

WWW.HACKINGARTICLES.IN

Contents

Introduction

Today we are discussing how to read hexadecimal bytes from an IP packet that helps a network admin identify various types of NMAP scanning. But before moving ahead, please read our previous articles, **"Network packet forensic"** and **"NMAP scanning with Wireshark".**

Requirement

Attacking Tool: Nmap
Analysis Tool: Wireshark

We are going to calculate the hexadecimal bytes of Wireshark using the given below table. As we know, Wireshark captures network packets mainly of 4 layers, which are described below in the table as per the OSI layer model and the 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 is familiar with basic NMAP scanning techniques; if not, read about it here. Now, open the terminal and run the "HOST SCAN" command to identify a live host in the network.

```
nmap -sn 192.168.1.100
```

Nmap performs host scans with the –sP/–sn flag and broadcasts ARP request packets to determine which IP address is assigned to the specific host machine.You can see that "1 host up" message in the image below.

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

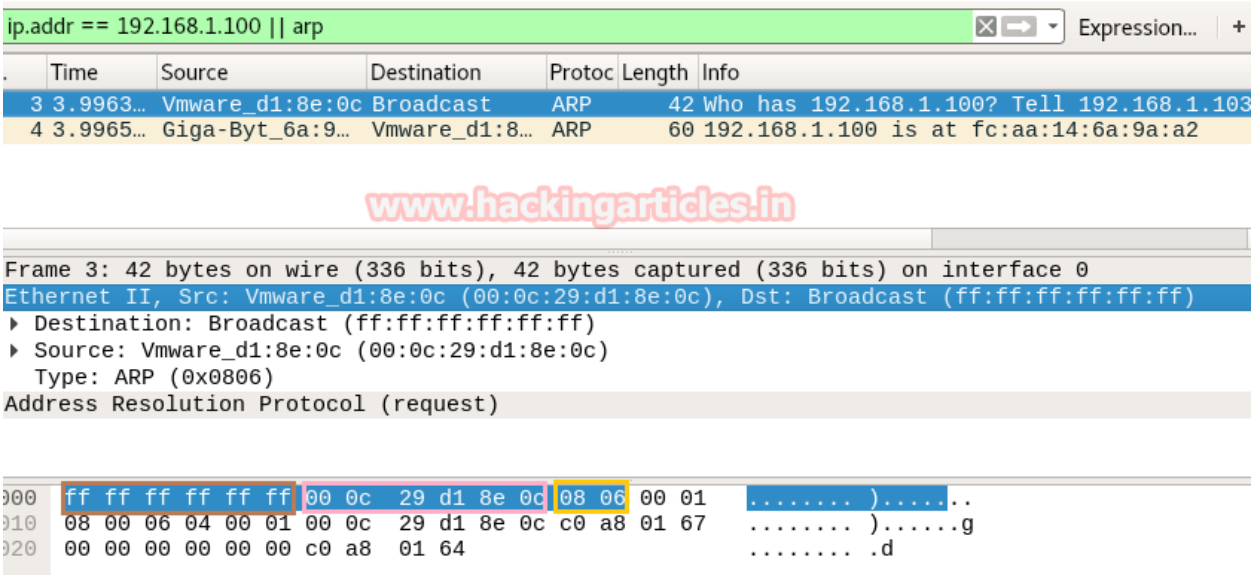
- **Collect Ethernet Header details**

In this case, we used Wireshark to capture network packets from the victim's network, and to analyse only the ARP packet, we used the filter **" ip.addr == VICTIM IP || arp,"** as shown in the image below. Here you will find 2 arp packets. Basically, the 1st arp packet is broadcasting IP to ask for the MAC address of that network, and the 2nd packet is unicast and contains the answer to the IP query.

Now let's read the hex value of the 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 what 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 the Ethernet header, we can conclude that it is an ARP broadcast packet asking for a destination Mac address. There shouldn't be any uncertainty in concern with the source Mac address of the person responsible for sending the packet, but if we talk about the destination Mac address, then we get ff:ff:ff:ff:ff:ff which means the exact destination is the machine that is not available here. Further moving ahead, we found **Ether type 0x0806** highlighted in yellow, which is used for the ARP protocol.

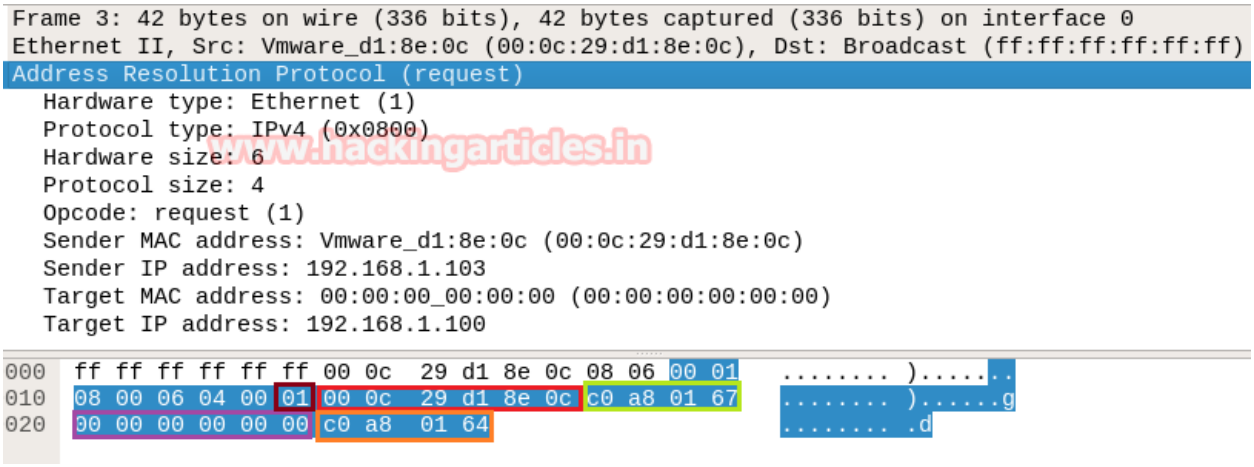


Collect ARP Header (Request/Reply)

In order to identify an ARP scan, you need to investigate some important parameters that could help a network admin make a correct assumption in regard to an ARP scan.

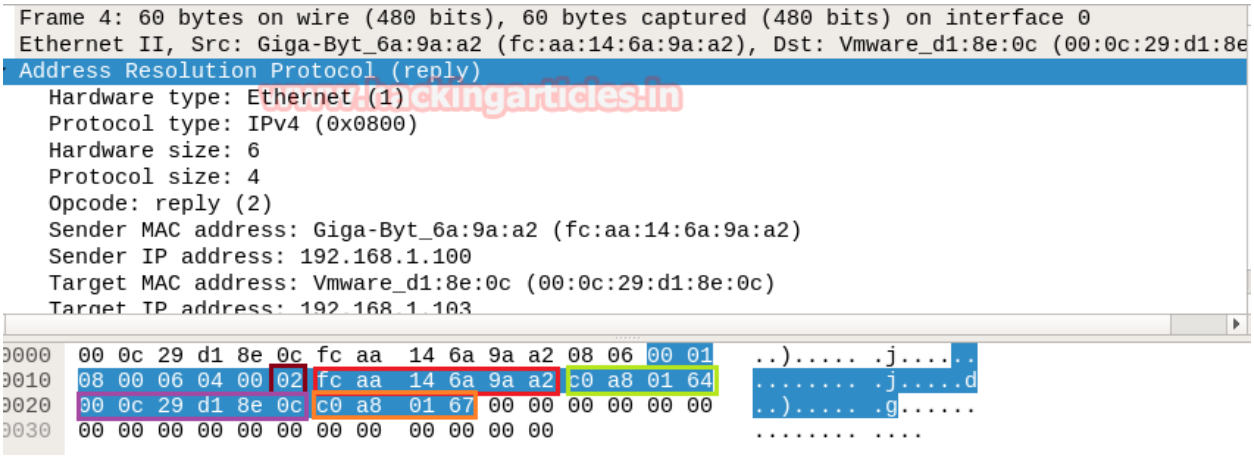
Try to collect the following details as given below:

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



With the help of the following table, you can read the hex value highlighted in the above and below images 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 run the "HOST SCAN" command to identify a live host in a network by sending a **Ping request** with the help of an 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 displays the "HOST UP" message shown in the image below.

```
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 the protocol version.**

For reading data from Ethernet heads, visit our previous article, **"Network packet forensic".**

NOTE: Ether type for IPv4 is 0x0800

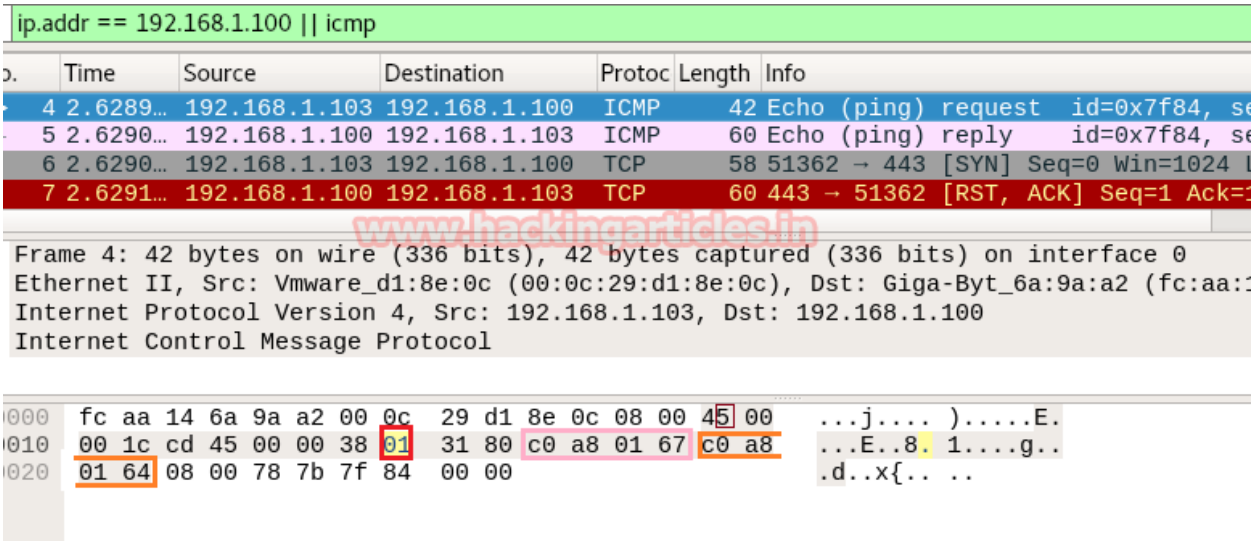
With the help of the IP header of a packet, since we know ICMP is a Layer 3 protocol according to the OSI model, we need to focus on the following details for ICMP forensics.

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

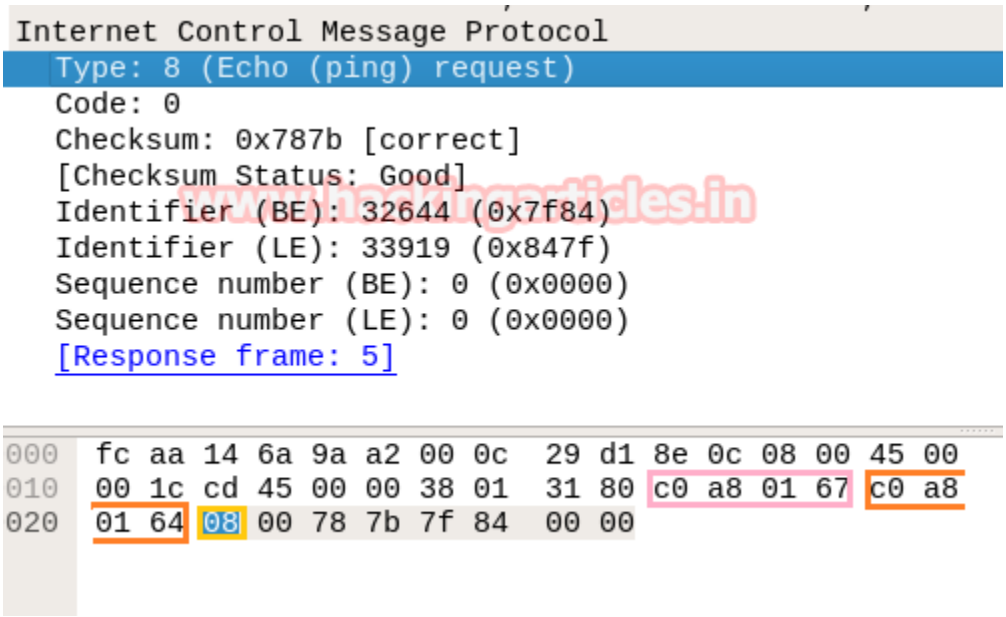
IP header (20 bytes)	**Header length**	**Protocol**	**Source IP**	**Destination IP**
Bits Color	**Brown**	**Red**	**Pink**	**Orange**
Hex Value	5	01	C0.a8.01.67	C0.a8.01.64
Decimal value	5	1	192.168.1.103	192.168.1.100



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)

As we discussed above, according to the Nmap ICMP scanning technique, the **1st packet** should be an **ICMP echo request** packet and the **2nd packet** should be an **ICMP echo reply** packet.



With the help of the following table, you can read the hex values highlighted in the above and below images 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


```
Internet Control Message Protocol
   Type: 0 (Echo (ping) reply)
   Code: 0
   Checksum: 0x807b [correct]
   [Checksum Status: Good]
   Identifier (BE): 32644 (0x7f84)
   Identifier (LE): 33919 (0x847f)
   Sequence number (BE): 0 (0x0000)
   Sequence number (LE): 0 (0x0000)
   [Request frame: 4]
   [Response time: 0.161 ms]
```

```
000   00 0c 29 d1 8e 0c fc aa   14 6a 9a a2 08 00 45 00
010   00 1c 66 c9 00 00 80 01   4f fc c0 a8 01 64 c0 a8
020   01 67 00 00 80 7b 7f 84   00 00 00 00 00 00 00 00
030   00 00 00 00 00 00 00 00   00 00 00 00
```

Identify TCP Flags

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

From the 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 a TCP SYN packet, then the bit for the **SYN flag** is set to **1**, for which the binary value will be **000000010** and its hexadecimal value will be **0x02.**

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

Sometimes you will get a combination of two or more flags in the TCP header, so in that scenario, take the help of the following table to read the hex value of such a packet to identify which TCP flag 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 designed the below table to let you know more about the 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** packets, from which we can calculate the source port and the destination port of the packet, respectively, as shown in the image 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.

Working of Default Scan for open port:

nmap -p80 192.168.1.100

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)

Gather IP Header Information for Protocol Version

For reading data from Ethernet heads, visit our previous article, **"Network packet forensic".**

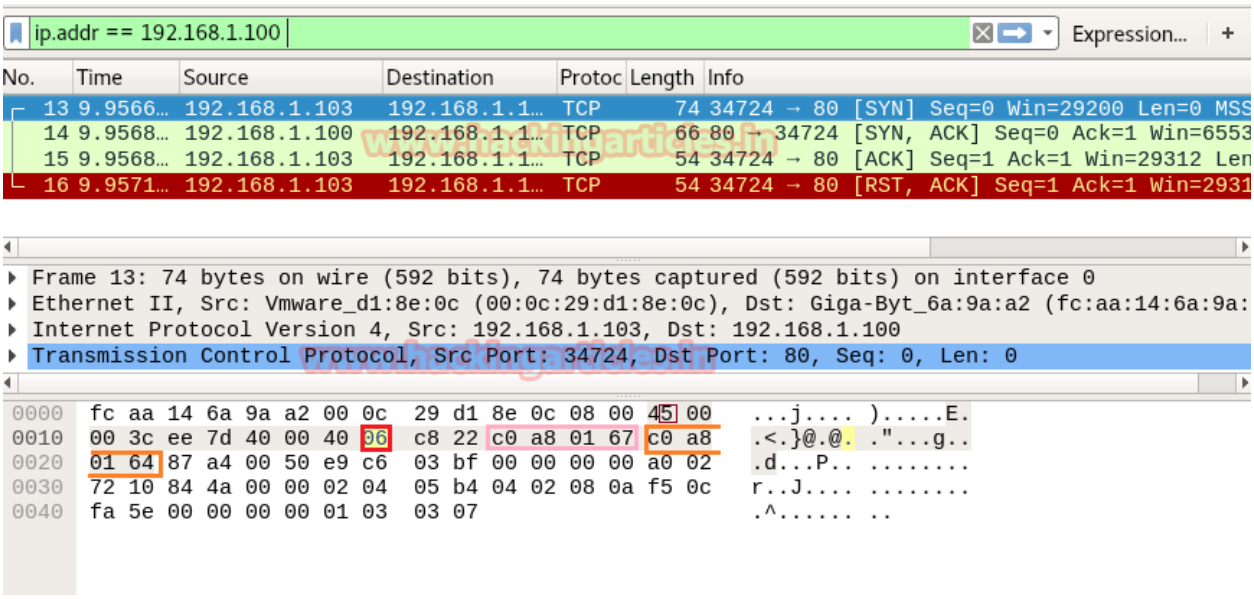
NOTE: Ether type for IPv4 is 0x0800.

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



Analysis TCP Header Details

From the above image, we had to obtain the 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 the source port in "light brown colour" and the destination port in "yellow colour". You can use the given 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 the next packet. Here we found that **hex value 12** indicates that **TCP-SYN/ACK** has been sent 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

Use the help given above to read the hex value of the given image. Hex value 12 for the TCP flag is used for SYN + ACK as explained above, and we get **0x12** by adding the hex values "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 the **TCP-RST** packet is used for sending a 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 a sequence gives an indication to the penetration tester that someone has chosen the NMAP default scan for network enumeration.

```
Transmission Control Protocol, Src Port: 38498, Dst Port: 80, Seq: 1, Len: 0
    Source Port: 38498
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 0
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x004 (RST)
    Window size value: 0
    [Calculated window size: 0]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x1daf [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
```



```
0000  fc aa 14 6a 9a a2 00 0c  29 d1 8e 0c 08 00 45 00   ...j.... ).....E.
0010  00 28 28 6a 40 00 40 06  8e 4a c0 a8 01 67 c0 a8   .((j@.@. .J...g..
0020  01 64 96 62 00 50 56 0b  21 58 00 00 00 00 50 04   .d.b.PV. !X....P.
0030  00 00 1d af 00 00                                  ......
```

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).
www.hackingarticles.in
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".**

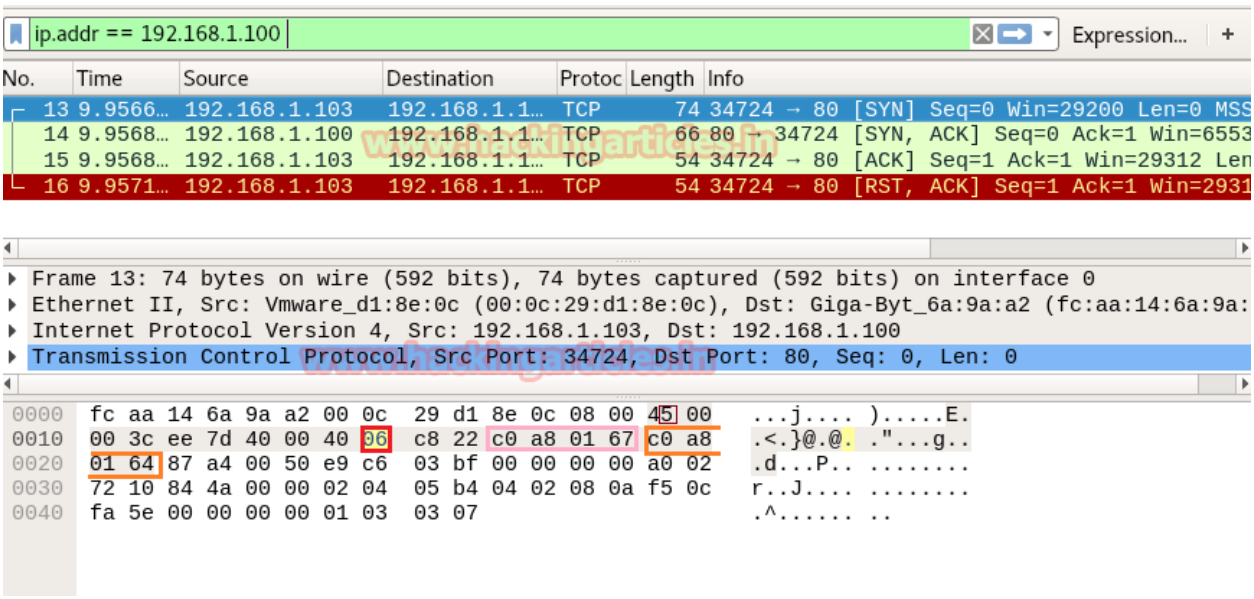
NOTE: Ether type for IPv4 is 0x0800.

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 the 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 the **TCP-SYN** packet is used for sending connection requests 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 the next packet. Here we found that 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                                             ..
```

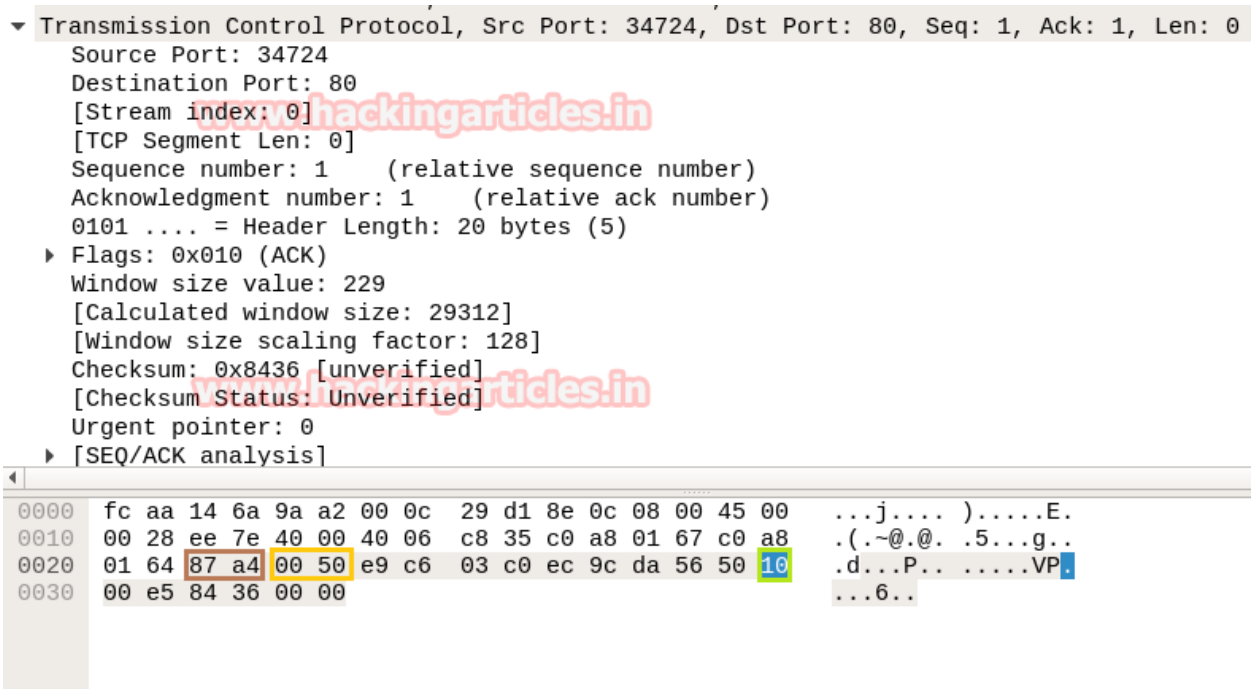
The only difference between Stealth Scan and TCP Scan is that here an ACK flag is sent by the source machine who initiated the TCP communication. Again, we read the next packet. Here we found that hex value 0x10 indicates that **TCP- ACK** has been sent via port 80.

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 the 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 a sequence gives an indication to the penetration tester that someone has chosen the NMAP default scan for network enumeration.

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



Nmap FIN Scan

In this case, we'll use TCP-FIN to enumerate the "OPEN" state of a specific port in any Linux-based system, so run the command below.

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

FIN's OperationScan for open ports: Send 2 packets of TCP-FIN to a specific port.
FIN is part of the TCP flag and NMAP uses the 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 from Ethernet heads, visit our previous article, "**Network packet forensic".**

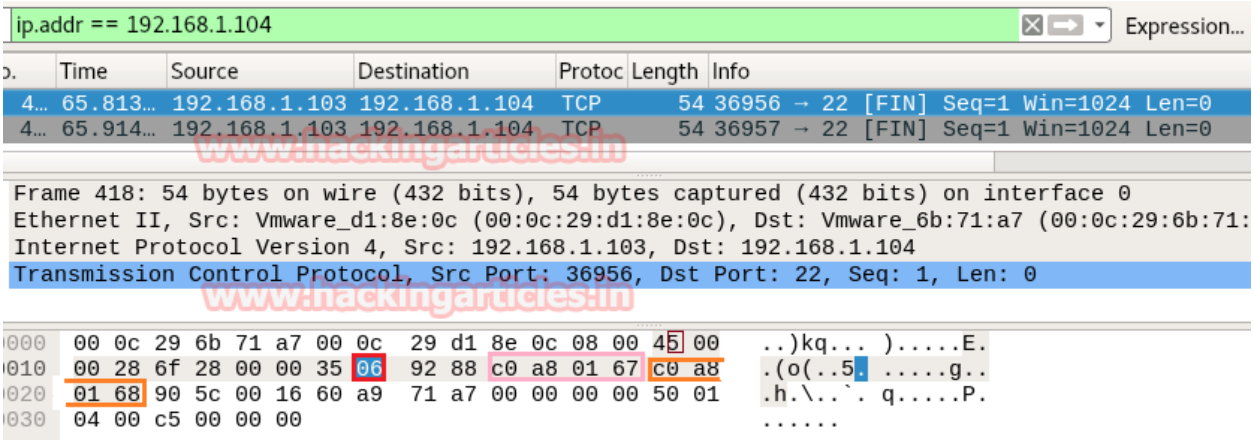
NOTE: Ether type for IPv4 is 0x0800

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

You can study these values using the table below to determine 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 let's identify the source and destination ports along with the flag hex value (TCP-FIN) so they are 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

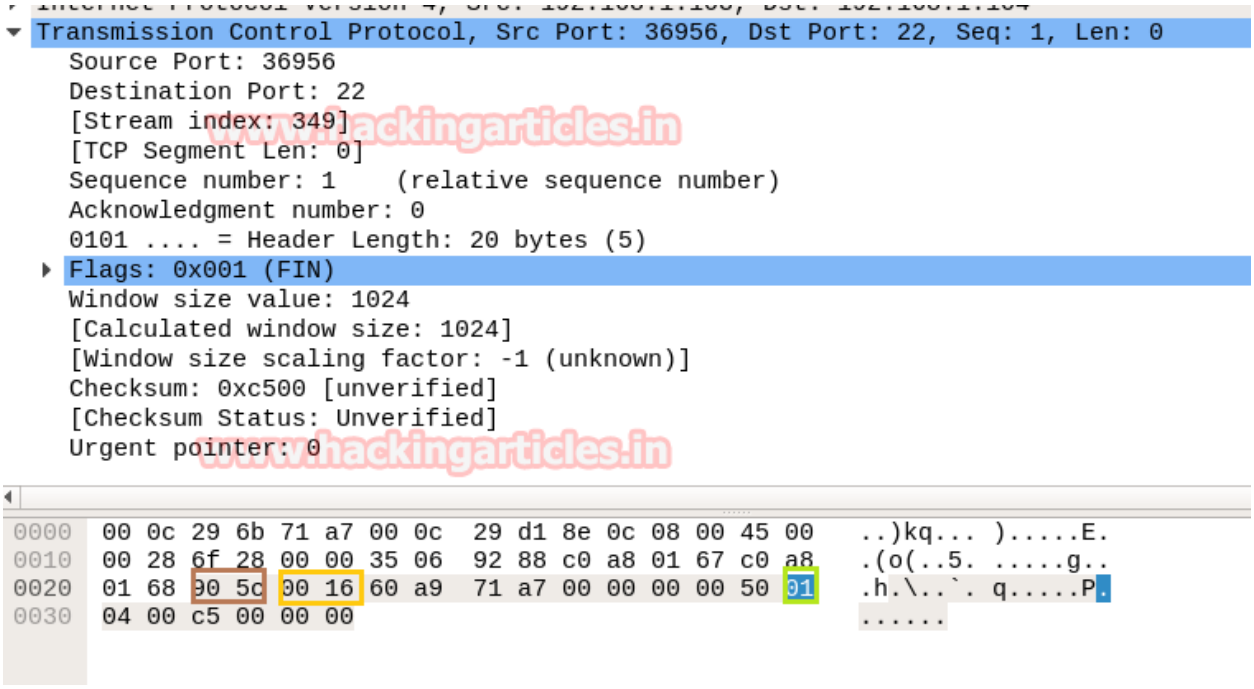
So, through the given below image and with the help of a table, we came to know that the TCP-FIN packet is used for sending connection requests on Port 22.

Conclusion:

So, as declared above regarding the working of the 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 a sequence gives an indication to the penetration tester that someone has chosen NMAP FIN scan for network enumeration.

NOTE: The presence of the first FIN packet (0x01) and the second RST packet (0x04) on the targeted network indicates a "Closed Port."



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
```

To use Null Scan for an open port, send two TCP-NONE packets to a specific port. Instead of using the three-way handshake protocol, NMAP used the NONE flag (No flag) to initiate TCP communication, and the bits of each flag were set to "0."

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

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-09 08:40 EST
Nmap scan report for 192.168.1.104
Host is up (0.00024s latency).

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


Step to Identify NMAP Null Scan

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

For reading data from Ethernet heads, visit our previous article, **"Network packet forensic".**

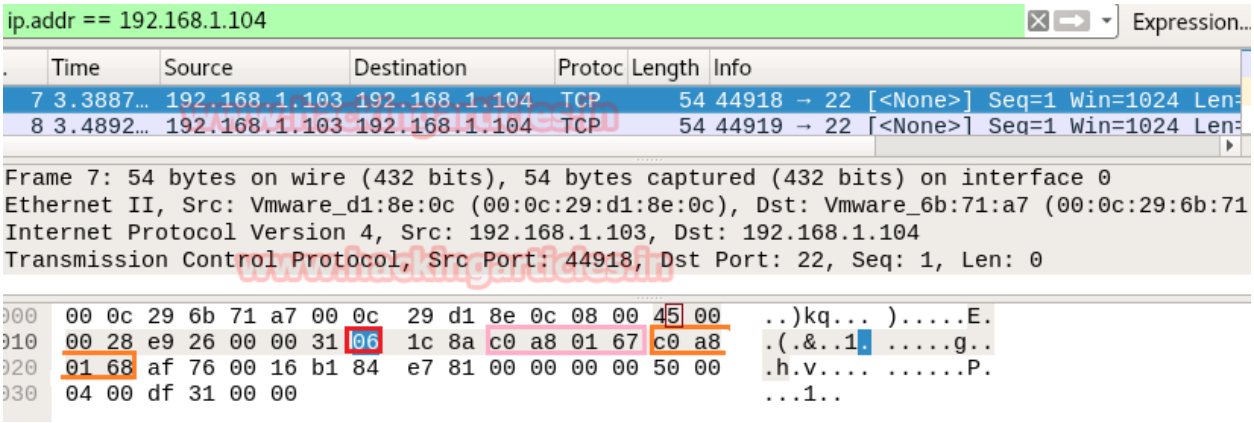
NOTE: Ether type for IPv4 is 0x0800

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

You can study these values using the provided table to determine 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 let's identify the source and destination ports along with the flag hex value **(TCP-NONE)** that is similar to 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 the given below image and with the help of a table, we come to know that here the TCP-NONE packet is used for sending connection requests on Port 22.

Conclusion:

So, as stated above regarding the working of the 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 a sequence gives an indication to the penetration tester that someone has chosen NMAP NONE scan for network enumeration.

NOTE: If you find the first NONE packet (0x00) and the second RST packet (0x04) on the target network, it indicates a "Closed Port."

```
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

In this case, we'll use the XMAS scan to list the "OPEN" state of any specific port in any Linux-based system.

nmap -sX -p22 192.168.1.104

Send **2 packets of TCP Flags** containing **FIN, PSH, and URG** on the specific port to perform an XMAS Scan for open ports.

Instead of following three-way handshake communications, NMAP used three TCP flags (FIN, PSH, and URG) to initiate TCP communication, with a bit of each flag set to "1."

```
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).

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 from Ethernet heads, visit our previous article, **"Network packet forensic".**

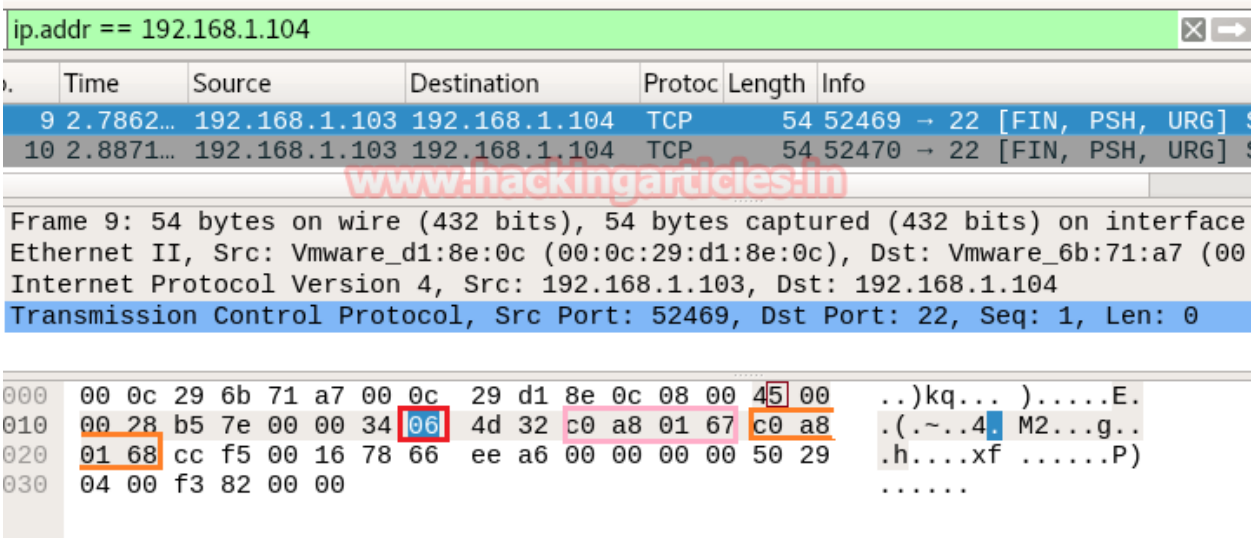
NOTE: Ether type for IPv4 is 0x0800

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 let's identify the source and destination ports along with the flag hex value (TCP-XMAS) 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 the given below image and with the help of the table, we come to know that here TCP flags {FIN, PSH, URG packets are used for sending connection requests on Port 22.

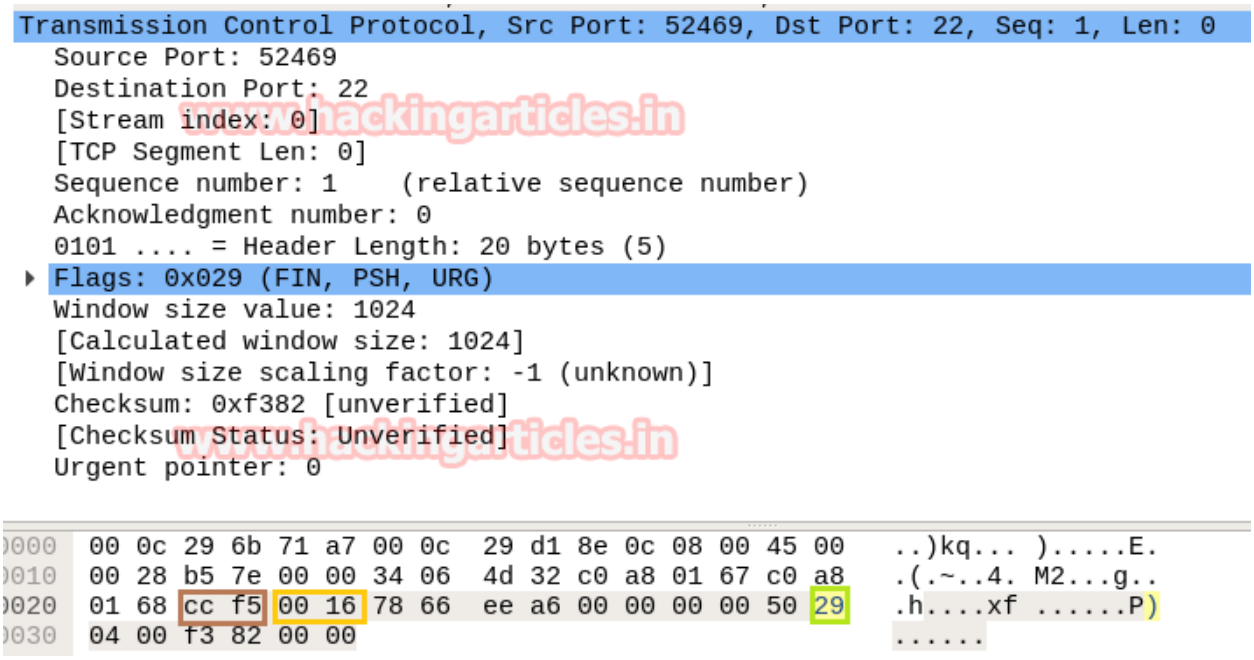
Conclusion!

So, as stated above regarding the working of the 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 a sequence gives the indication to the penetration tester that someone has chosen NMAP XMAS scanned for network enumeration.

NOTE:

- **If you discovered the first FIN, PSH, or URG packet (0x29) and the second RST packet (0x04) on the targeted network, indicate "Closed Port.**
- **"NMAP FIN, NMAP NULL, and NMAP XMAS scans are only applicable on Linux-based systems.**



Nmap UDP Scan

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

```
nmap -sU -p 68 192.168.1.104
```

The operation of the XMAS Scan for open ports is as follows: Send **2 packets of UDP** to a specific port.

It is quite different from the TCP communication process in that here no flag is used for establishing a connection or initiating 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).
www.hackingarticles.in
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 from Ethernet heads, visit 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 very similar to the NMAP above scan in that the "IP header" and "Ethernet header" information will be the same whether it is TCP communication or UDP communication, and you can study these values to obtain their original value using the provided table.

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 the hex value used for the UDP protocol, which is quite useful in identifying NMAP UDP scans from remanding scanning methods.

```
  7 1.3272… 192.168.1.103 192.168.1.104  UDP      42 33397 → 68 Len=0
  8 1.4279… 192.168.1.103 192.168.1.104  UDP      42 33398 → 68 Len=0
```

```
Frame 7: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on inter
Ethernet II, Src: Vmware_d1:8e:0c (00:0c:29:d1:8e:0c), Dst: Vmware_6b:71:a7
Internet Protocol Version 4, Src: 192.168.1.103, Dst: 192.168.1.104
User Datagram Protocol, Src Port: 33397, Dst Port: 68
```

```
000  00 0c 29 6b 71 a7 00 0c  29 d1 8e 0c 08 00 45 00   ..)kq... ).....E.
010  00 1c 15 d3 00 00 2c 11  f4 de c0 a8 01 67 c0 a8   ......,. .....g..
020  01 68 82 75 00 44 00 08  f9 04                     .h.u.D.. ..
```

Analysis UDP Header Details

Now let's identify the source and destination ports, 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 a sequence gives the penetration tester an indication that someone has chosen NMAP UDP scan for network enumeration.

NOTE: If the first UDP packet and the second UDP with an ICMP Message Port are both unreachable, it indicates that the target network has a "Closed Port."

```
User Datagram Protocol, Src Port: 33397, Dst Port: 68
    Source Port: 33397
    Destination Port: 68
    Length: 8
    Checksum: 0xf904 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
```

```
000  00 0c 29 6b 71 a7 00 0c  29 d1 8e 0c 08 00 45 00   ..)kq... ).....E.
010  00 1c 15 d3 00 00 2c 11  f4 de c0 a8 01 67 c0 a8   ......,. .....g..
020  01 68 82 75 00 44 00 08  f9 04                     .h.u.D.. ..
```

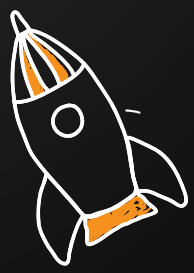
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

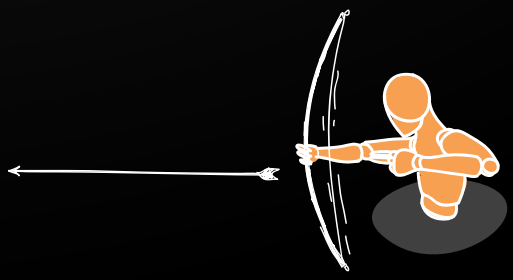
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux