

# **Forensic Investigation**

of Nmap Scan with

# Wireshark

WWW.HACKINGARTICLES.IN

# Contents

Introduction	3
Requirement	3
Nmap ARP Scanning	3
Step to Identify Nmap ARP Scan	4
Nmap ICMP Scanning	6
Step to Identify NMAP ICMP Scan	7
Identify TCP Flags	9
Default NMAP Scan (Stealth Scan)	11
Step to Identify NMAP Default Scan (Stealth Scan)	12
Analysis TCP Header Details	13
Nmap TCP Scan	16
Step to Identify NMAP TCP Scan	16
Nmap FIN Scan	19
Step to Identify NMAP FIN Scan	20
Analysis TCP Header Details	21
Nmap NULL Scan	21
Step to Identify NMAP Null Scan	22
Analysis TCP Header Details	23
Nmap XMAS Scan	24
Step to Identify NMAP XMAS Scan	24
Nmap UDP Scan	26
Step to Identify NMAP UDP Scan	27
Analysis UDP Header Details	28



# 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	Destination MAC Address	Source MAC Address	Ether Type
14 bytes	6 Bytes	6 Bytes	2 Bytes
	www.uaeking	arddestin	
Bits Color	Brown	Pink	Yellow
Hexadecimal value	ff:ff:ff:ff:ff:ff:ff	00:0c:29:d1:8e:0c	0806



i	ip.addr == 192.168.1.100    arp Expr												
	Time	Source	Destination	Protoc Lengt	n Info								
	3 3.9963	Vmware_d1:8e:0c	Broadcast	ARP 4	2 Who ha	as 192.168.1.	100? Tell	192.168.1	.103				
	4 3.9965	Giga-Byt_6a:9…	Vmware_d1:8	ARP 6	0 192.16	68.1.100 is a	at fc:aa:14	4:6a:9a:a2					

# www.hackingarticles.in

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)
Destination: Broadcast (ff:ff:ff:ff:ff)
▶ Source: Vmware_d1:8e:0c (00:0c:29:d1:8e:0c)
Type: ARP (0x0806)
Address Resolution Protocol (request)

000	ff	ff	ff	ff	ff	ff	00	0c	29	d1	8e	0c	08	06	00	01	 )
910	08	00	06	04	00	01	00	0c	29	d1	8e	0c	c0	a8	01	67	 )g
920	00	00	00	00	00	00	c0	a8	01	64							 .d

#### **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
1	www	hackingar	ides in		
Decimal value of Request	1	Noneed	192.168.1.103	Noneed	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	Noneed	192.168.1.100	Noneed	192.168.1.103

Frame 4: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0 Ethernet II, Src: Giga-Byt\_6a:9a:a2 (fc:aa:14:6a:9a:a2), Dst: Vmware\_d1:8e:0c (00:0c:29:d1:8e Address Resolution Protocol (reply) Hardware type: Ethernet (1) Protocol type: IPv4 (0x0800) Hardware size: 6 Protocol size: 4 Opcode: reply (2) Sender MAC address: Giga-Byt\_6a:9a:a2 (fc:aa:14:6a:9a:a2) Sender IP address: 192.168.1.100 Target MAC address: Vmware\_d1:8e:0c (00:0c:29:d1:8e:0c) Target TP address: 192,168,1,103 0000 00 0c 29 d1 8e<u>0c</u>fc aa 14 6a 9a a2 <u>08 06 00 01</u> . . ) . . . . . 08 00 06 04 00 02 fc aa 14 6a 9a a2 c0 a8 01 64 00 0c 29 d1 8e 0c c0 a8 01 67 00 00 00 00 00 00 9010 
 00
 0C
 29
 d1
 8e
 0C
 c0
 a8
 01
 67
 00
 00

 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00</ 9020 . . . . . . . . . .

# 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 hackinga	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



	ip.addr == 192.168.1.100    icmp													
р.		Time	Source		Destinat	on	Protoc	Length	Info					
	4	2.6289	192.168	3.1.103	192.16	8.1.100	D ICMP	42	Echo	(ping)	reques	st id=0x7	'f84, se	
-	5	2.6290	192.168	3.1.100	192.16	8.1.103	3 ICMP	60	Echo	(ping)	reply	id=0x7	'f84, se	
	6	2.6290	192.168	3.1.103	192.16	8.1.100	) TCP	58	51362	2 → 443	[SYN]	Seq=0 Wir	n=1024 l	
	- 7	2.6291	192.168	3.1.100	192.16	8.1.103	3 TCP	60	443 -	→ 51362	[RST,	ACK] Seq=	-1 Ack=1	
				0	mmm2	had	haar	Halac	affa					
I	Frai	me 4: 42	bytes	on wire	e (336 b	its), 4	42 byte	s captu	ured	(336 bi	ts) on	interface	e 0	
ł	Eth	ernet II	, Src:	Vmware_	d1:8e:0	c (00:0	0c:29:d	1:8e:00	c), D:	st: Gig	a-Byt_(	6a:9a:a2 (	fc:aa::	
1	Int	ernet Pr	otocol	Versior	1 4, Sro	: 192.:	168.1.1	03, Dst	t: 192	2.168.1	.100			
1	Int	ernet Co	ntrol M	essage	Protoco	1								
				-										
0	~ ~	<i>.</i>				4.0			~		`	-		
0	00	TC aa 1	4 6a 9a	i a2 00	<u>0C</u> 29	d1 8e	0C 08 0	0 45 0	Θ.	]	)	.E.		

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.

.d..x{...

#### Identify ICMP Message type (Request /Reply)

01 64 08 00 78 7b 7f 84 00 00

020

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.

```
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
    01 64 08 00 78 7b 7f 84
020
                              00 00
```

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.



IPHeader =>	ІСМР Туре	Source IP	Destination IP
Bits color	Yellow	Pink	Orange
ICMP Echo Request Hex Value	108aCKIng	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

T	ype:	0	(Ec	ho	(pi	.ng)	re	eply	/)							
С	ode:	Θ														
С	heck	sum	n: 0	x80	7b	[co	rre	ect]								
្រ	Chec	ksu	Im S	tat	us:	Go	od	TET	- AP	f	a ft	5				
Ī	dent	ifi	er	(BE	):	326	644	(0x	7f84	1)		U				
T	dent	ifi	er	(LE	:):	339	19	(0x	(847f	÷						
S	eane	nce	nu	mbe	r (	BF)	: 0	) (6	)x000	) ຄົງ						
s	enne	nce	nu	mhe	r (	IE)			x000	00)						
r	Degu	est	fr	ame		1		, (0	//////	,0,						
- L	Docn	one		ime		16	1 n	ne l								
L	Resp	ons	e i	TILE		. 10	л II	12]								
000	00	0c	29	d1	8e	Θс	fc	aa	14	6a	9a	a2	08	00	45	00
010	00	1c	66	с9	00	00	80	01	4f	fc	сO	a8	01	64	c0	a8
920	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	АСК	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	hack <sup>32</sup> marf	<b>a</b> 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.
910	00	2c	fa	3e	00	00	33	06	09	72	c0	a8	01	67	сO	a8	.,.>3.	.rg
920	01	64	c8	a2	01	bb	bc	af	75	68	00	00	00	00	60	02	.dι	uh`.
930	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 Clestin	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,

							nB	nord	D-A-	YT	ard?	ß		Bra			
000	00	0c	29	d1	8e	0c	fc	aa	14	6a	9a	a2	08	00	45	00	)jE.
)10	00	28	66	ca	40	00	80	06	0f	ea	c0	a8	01	64	сO	a8	.(f.@d
)20	01	67	01	bb	c8	a2	00	00	00	00	bc	af	75	69	50	14	.guiP <mark>.</mark>
020	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 <u>I</u>P 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	Header	Protocol	Source IP	Destination IP		
(20 bytes)	length	nakihmari	fides.fn			
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.



ļ	p.ac	ldr =	= 19	2.1	58.1	.100	)																Þ	<] -	•	Exp	ress	ion	.	+
No.		Time	,	S	ourc	e			0	Destin	atio	n		Prot	oc L	engt	h In	fo												
	13 14 15 16	9.9 9.9 9.9 9.9	566. 568. 568. 571.	. 1 . 1 . 1	92. 92. 92. 92.	168 168 168 168	.1. .1. .1. .1.	103 100 103 <mark>103</mark>	W	192.1 192.1 192.1 192.1	168 168 168 168			TCP TCP TCP TCP	E		743 668 543 543	4724 0 → 4724 4724	1 → 347 1 → 1 →	80 24 80 80	[SYN [SYN [ACK [RST		Seq= ACK] Seq= ACK]	0 k Se 1 A Se	/in=2 eq=0 kck=: eq=1	2920 Ack 1 Wi Ack	)0 (=1 [n=2 (=1	Len= Wir 2931 Wir	=0 1=6 L2 1=2	1SS 553 Len 931
<ul> <li>F</li> <li>F</li></ul>	ra th	ne 1 erne erne	3: et I et P	74 I, rot	byt Src oco	es : V l V	on /mwa /ers	wir re_ ion	e ( d1:	592 8e:0 Src	bit )c (	s), 00: 92.	74 0c: 168	by 29:	tes d1: 103	ca 8e: , D	ptu 0c), st:	red , Ds 192	(59 t: 2.16	2 b Gig 8.1	its) a-By .100	or t_f	n int Sa:9a	ter a:a	face 2 (1	e 0 fc:a	a:1	L4:6	ia:9	)a:
4	Tu			on	con		-			4/ 9	Let .	, M	غان	<b>N</b> HI	677	03	4 1 1		00	,	cy.	۰,	Lon							Þ
	00 10 20 30 40	fc 00 01 72 fa	aa 3c 64 10 5e	14 ee 87 84 00	6a 7d a4 4a 00	9a 40 00 00	a2 00 50 00 00	00 40 e9 02 01	0c 06 04 03	29 c8 03 05 03	d1 22 bf b4 07	8e 00 04	0c a8 00 02	08 01 00 08	00 67 00 0a	4 <mark>5</mark> a0 f5	00 a8 02 0c	.( r	j. <.}@ d J. ^	).@. .P.	)		.E. g							

# **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-B-D	Green		
TCP-SYN Packets Hex value	92 62	00 50 10 010 010 51	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
)000 fc aa 14 6a 9a a2 00 0c 29 d1 8e 0c 08 00 45 00
                                                         ...j... )....E.
010 00 2c ea 8e 00 00 38 06 14 22 c0 a8 01 67 c0 a8
                                                         .,...8. ."...g..
020 01 64 96 62 00 50 56 0b 21 57 00 00 00 00 60 02
                                                         .d.b.PV. !W....`.
030 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	100 50 Gleking	92 62 CLESHIN	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]
)000 00 0c 29 d1 8e 0c fc aa  14 6a 9a a2 08 00 45 00
                                                         ..).... .j....E.
)010 00 2c <u>69 27</u> 40 00 80 06 0d 89 c0 a8 01 64 c0 <u>a8</u>
                                                         .,i'@... ....d..
020 01 67 00 50 96 62 17 52 e1 dc 56 0b 21 58 60 12
                                                        .g.P.b.R ..V.!X`.
030 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 Coro Bro	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, Seg: 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 <u>28 6a</u> <u>40 00</u> 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).
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	Header	Protocol	Source IP	Destination IP		
(20 bytes)	length	addhraat	ildes.in			
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.

📕 ip	.addr == 19	2.168.1.100			Expression +
No.	Time	Source	Destination	Protoc Length Info	
	3 9.9566. 4 9.9568. 5 9.9568. 6 9.9571.	192.168.1.1 192.168.1.1 192.168.1.1 192.168.1.1	03         192.168.1.1           00         192.168.1.1           03         192.168.1.1           03         192.168.1.1	. TCP 74 34724 → 80 . TCP 66 80 → 34724 TCP 54 34724 → 80 TCP 54 34724 → 80	[SYN] Seq=0 Win=29200 Len=0 MSS [SYN, ACK] Seq=0 Ack=1 Win=6553 [ACK] Seq=1 Ack=1 Win=29312 Len [RST, ACK] Seq=1 Ack=1 Win=2931
<ul> <li>F</li> <li>F</li> <li>F</li> <li>F</li> <li>F</li> <li>F</li> <li>F</li> </ul>	rame 13: Thernet I Internet P Tansmissio	74 bytes on w I, Src: Vmwar rotocol Versi <mark>on Control P</mark> r	ire (592 bits), e_d1:8e:0c (00:0 on 4, Src: 192.1 otocol, Src Port	74 bytes captured (592   c:29:d1:8e:0c), Dst: Gi 68.1.103, Dst: 192.168. : 34724, Dst Port: 80, 9	pits) on interface 0 ga-Byt_6a:9a:a2 (fc:aa:14:6a:9a: 1.100 Seq: 0, Len: 0
000 001 002 003 004	<ul> <li>fc aa</li> <li>00 3c</li> <li>01 64</li> <li>72 10</li> <li>fa 5e</li> </ul>	14 6a 9a a2 ( ee 7d 40 00 4 87 a4 00 50 6 84 4a 00 00 ( 00 00 00 00 00 (	00       0c       29       d1       8e       0         40       06       c8       22       c0       a         e9       c6       03       bf       00       0         02       04       05       b4       04       0         01       03       03       07       0	IC 08 00 4500j 8 01 67 C0 a8 .<.}@.@ 10 00 00 a0 02 .dP. 12 08 0a f5 0c rJ .^	. )E. "g 

#### • 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 estin	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.



 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
 ...j...)...E.

 0020
 01 64 87 a4 00 50 e9 c6
 03 bf 00 00 00 00 a0 02
 ...j....

 01 64 87 a4 00 00 02 04
 05 b4 04 02 08 0a f5 0c
 ...j....

 020
 72 10 84 4a 00 00 02 04
 05 b4 04 02 08 0a f5 0c
 r...j...

 040
 fa 5e 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 20 00 00 00	87 a4	12
Decimal Value	80	34724	18

•	Tra	ansmi	ssi	on	Con	tro	1 P	rot	ocol,	Sro	: Po	rt:	80,	Ds	tΡ	ort:	34724,	Seq	: 0,	Ack:	1,	Len:
		Sourc	e P	ort	: 8	0																
		Desti	nat	ion	Po	rt:	34	724														
		[Stre	am	ind	ex:	0]	o B		0.9-2		<b>19</b> -	floor	h Arr	· · ·								
		[TCP	Seg	men	t L	en:	0]	C.C	ьщ	للعرو	ul		ш									
		Seque	ence	nu	mbe	r:	0	(	relat	ive	sec	quen	ce r	numb	er)	)						
		Ackno	wle	dgm	ent	nu	mbe	r:	1	(re	lati	Lve	ack	nur	ıber	-)						
		1000		. =	Не	ade	r L	eng	th: 3	2 by	/tes	s (8	)									
	•	Flags	s: 0	x01	.2 (	SYN	I, A	CK)														
	1	Windo	w s	ize	va	lue	: 6	553	5													
		[Calc	ula	ted	wi	ndo	ws	ize	: 655	35]												
	(	Check	sum	1: 0	xae	76	[un	ver	ified	ן ו												
		[Chec	ksu	m S	tat	us:	Un	ver	ified	100-		- <b>A</b> -										
		Ürger	nt p	oin	ter	: 0	30	KI	ાઇલ	<b>LUC</b>	16F	ш	J									
	•	Optic	ns:	(1	2 b	yte	s),	Ма	ximun	n se	amer	nt s	ize,	No	- Op	erat:	ion (NO	Р),	Wind	ow sc	ale	, No-
	•	[SEQ/	ACK	àn	aly	sis	11				-		,		•							, 
•															,							
$\odot$	000	00	0c	29	d1	8e	0c	fc	aa :	14 6	a 9a	a a2	08	00	45	00	)		j	.E.		
0	010	00	34	52	33	40	00	80	06 2	24 7	5 c(	9 a8	01	64	c0	a8	.4R3@.	\$	Su	d		
0	020	01	67	90	50	87	a4	ec	9c (	da 5	5 e	9 c6	03	c0	80	12	.g.P		υ			
0	030	ff	ff	ae	76	00	00	02	04 (	95 b	4 0:	1 03	03	08	01	01	v					
$\odot$	040	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	<b>Destination Port</b>	Hex value of Flag
Bits Color	Light Brown	Yellow Doo fto	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	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

ip.	addr =	= 19	2.1	58.1.	.104																			$\times$	-	<u> </u>	Expre	ssion
o.	Time	•	S	ource	e			Des	tinati	on		F	Proto	be L	ength	In	fo											
4.	. 65.	813.	. 1	92.3	168	.1.:	103	192	2.16	8.1	.104	4 .	ТСР		54	43	6956		22	[FI	١]	Sec	<b>1=1</b>	Wi	n=1	024	Lei	า=0
4	. 65.	914.	. 1	92.:	168	.1.6	103	192	2.16	3.1	. 104	<b>h</b>	ТСР	2	54	43	6957		22	[FII	١]	Sec	1=1	Wi	n=1	024	Lei	า=0
				w	u.	a	ue.		щe	9	and	100		U														
Fr	ame 4	18:	54	by	tes	on	wi	re	(432	bi	ts)	, 5	4 b	yte	s ca	ιpt	ured	(4	132	bits	s)	on	int	ter	fac	e 0	)	
Et	herne	et I	I,	Src	: V	mwa	re_	d1:	8e:0	с (	00:	0c:	29:	d1:	8e:0	)c)	, Ds	t:	Vmw	are_	_6b	:71	L:a7	7 (	00:	0c:	29:	6b:71:
In	terne	et P	rot	oco	1 V	ers	ion	4,	Src	: 1	92.	168	.1.	103	, Ds	st:	192	.16	58.1	104	4							
Tr	ansmi	.ssi	on	Con	tro	ΙP	rot	oco.	L, S	rc	Por	t i	369	56,	Dst	:Р	ort:	22	2, S	seq:	1,	Le	en:	0				
				u	<sup>1</sup>		LLC:	19r	шę	للنكر			50	LU,														
000	00	0c	29	6b	71	a7	00	Θc	29	d1	8e	0c	08	00	45	00		.)k	q	. ).		E						
010	00	28	6f	28	00	00	35	06	92	88	сO	a8	01	67	C0	a8		(0(	5			.g.						
020	01	68	90	5c	00	16	60	a9	71	a7	00	00	00	00	50	01	. 1	ı.\	••`	. q.	•••	Р	•					
030	04	00	c5	00	00	00													••									



# **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	0016	01 lanafia
Decimal Value	36956	22	harderenn

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."



# 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	Header length	Protocol	Source IP	Destination IP
(20 bytes) 🛛 🕦	ww.hack	ingartic	es-in	
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



ip.a	dr ==	: 192.1	168.1	.104	ŀ																	×	- 0	Expre	ession
	Time	9	Sourc	e			Des	tinati	on		F	Proto	c Le	ngth	Info										
7	3.38	87 :	192.	168	<b>.</b> 1n	103	192	.168	3.1	.104		ТСР	<b>_</b>	54	449	18 -	→ 22	[<	Non	e>]	Seq	=1	Win=	1024	Len
8	3.48	92 :	192.	168	.1.:	103	192	. 168	3.1	.104		TCP	J	54	449	19 -	→ 22	[<	Non	e>]	Seq	=1	Win=	1024	Len
Fra Eth Int Tra	me 7: ernet ernet nsmis	: 54 : II, : Pro ssion	byte Src toco Con	s o : V l V tro	n w mwa ers 1 P	ire re_ ion rot	(43 d1:8 4, oco]	32 b 3e:0 Src L, S	its c ( : 1 rc	), 00: 92. Por	54 9c: 168	byte 29:0 .1.: 449:	es ( d1:8 103, 18,	captu Be:00 , Dst Dst	ured c), t: 1 Por	(43 Dst 92.3 t: 2	32 b: : Vm 168.: 22, \$	its war 1.1 Seq	) o e_6 .04  : 1	n in b:7: , Le	nter 1:a7 en:	fac (0	:e 0 )0:00	:29:	6b:71
000	00	0c 29	9 6b	71	a7	00	0c	29	d1	8e	0c	08	00	45 0	0	)	ka		)	E					
010	00	28 e9	26	00	00	31	06	1c	8a	сØ	a8	01	67	c0 a	8	.(.	&1			.g.					
020	01	68 af	76	00	16	b1	84	e7	81	00	00	00	00	50 0	Θ	.h.	V	_		. P					

...1..

# 030 04 00 df 31 00 00

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	0016	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

# Nmap XMAS Scan

030

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

04 00 df 31 00 00

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
Mmap 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)
Wmap 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



.h.v....P.

...1..

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	Header length	Protocol	Source IP	Destination IP
(20 bytes) 🛛 🚺	ww.hack	ingartic	es:in	
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

ip.a	ddr =	= 192	2.168	.1.104	4																$\times$	•
).	Time		Sou	rce			Dest	tinati	on			Proto	c L	ength	Inf	o						
	2.78	362	192	168	.1.	103	192	.16	8.1	.10	4	TCP		54	52	469		22	[FIN,	PSH,	URG]	
16	92.80	5/1	192	108	.1.	103	192	.10	8.1	.10	4	TCP	d	54	52	2470	→	22	[FIN'	, PSH,	UKG]	
Fra	ame 9	: 54	byt	tes o	on w	ire	(43	32 b	its	),	54	byt	es	capt	ure	ed (4	432	bi	ts) (	on int	erfac	e
Eth	nerne	t II	., Sı	rc: V	/mwa	re_	d1:8	Be:0	с (	00:	0c:	29:	d1:	8e:0	c),	Ds	t:	Vmw	are_(	6b:71	a7 (0	00
Int	terne ansmi	t Pr ssio	otoc n Co	o⊥ V ontro	/ers	10n rot	4, ocol	Src	: 1 rc	.92. Por	168 t:	524	103 69.	, Ds Dst	t: Po	192 rt:	.16	8.1 . S	.104 Seg: 1	I. Lei	n: 0	
		0010						-, -				021	,	2000				, -		2, 20,		
000	00	0c 2	29 6	b 71	a7	00	0c	29	d1	8e	0c	08	00	45 0	0		)kc		. )	E.		_
010	00	28	b5 7	e 00	00	34	<u>06</u>	4d	32	00	a8	01	67	C0 a	18	. (	.~.	.4	. M2.	g		

#### • Analysis TCP Header Details

030 04 00 f3 82 00 00

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.



0000	00	0c	29	6b	71	a7	00	0c	29	d1	8e	0c	08	00	45	00	)kq	)E.
010	00	28	b5	7e	00	00	34	06	4d	32	c0	a8	01	67	сO	a8	.(.~4.	M2g
020	01	68	сс	f5	00	16	78	66	ee	a6	00	00	00	00	50	29	.hxf	P)
020	04	00	†3	82	00	00										_		

## **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).
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	Header length	Protocol	Source IP	Destination IP		
(20 bytes)	www.had	kingarti	des in			
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."

Usei	- Da	tag	ram	Pr	oto	col	., S	rc	Port:	: 3	339	7,	Dst	Po	rt:	68	
S	Source Port: 33397																
D	Destination Port: 68																
L	Length: 8																
С	Checksum: 0xf904 [unverified]																
[	[Checksum Status: Unverified]																
Ē	Stre	am	ind	lex:	1]												
						-											
0000	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	<u>,</u> g
000	~ 4	~ ~	0.0		00		~~	~~		$\sim 1$							





# JOIN OUR TRAINING PROGRAMS



in 😱

www.ignitetechnologies.in