Hacking Articles

Raj Chandel's Blog

Menu

A Home » Nmap » Forensic Investigation of Nmap Scan using Wireshark

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



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: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: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.

i	o.addr == 192	.168.1.100 arp								Expre	ssion	+
	Time	Source	Destination	Protoc	Length	Info						
	3 3.9963	Vmware_d1:8e:0c	Broadcast	ARP	42	Who	has	192.168.1.	100? Tel	192.	168.1	.103
	4 3.9965	Giga-Byt_6a:9…	Vmware_d1:8	ARP	60	192.	168.	1.100 is a	t fc:aa::	4:6a:	9a:a2	
			mmmbed	h	artha		fra					

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 Destination: Broadcast (ff: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	· · · · · · · ·) · · · · · · ·
010	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 ARP scan, you need to investigate some important parameters whic^L ≈ 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
                                                                      )....
     08 00 06 04 00 <mark>01 00 0c</mark>
00 00 00 00 00 00 c0 a8
                                29 d1 8e 0c c0 a8 01 67
010
020
                               01 64
```

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
	www	hackingar	tidles 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 Ether	4 net	: 6 : I	0 b I,	yte Src	s o : G	n w iga	ire -By	(48 t_6a	30 b a:9a	its :a2), (f	60 c:a	byt a:1	es 4:6	cap a:9	ture a:a2	ed (480 2), Dst:	bits) Vmwa	on are_c	inter 1:8e:	fac 0c	ce 0 (00:0c	:29:d1	:8e
Addre	SS	Re	sol	uti	on	Pro	toc	ol (rep	ly)		_			•									
Har	dwa	are	ty	pe:	Et	her	net	(1) CK	(III)	ga	Бe	e	es	-III									
Pro	to	col	ty	pe:	IP	v4	(0x	080	Ð)															
Har	dwa	are	si	ze:	6																			
Pro	to	col	si	ze:	4																			
0pc	od	e:	rep	ly	(2)																			
Sen	Ide	гM	AC	add	res	s:	Gig	a-By	/t_6	a:9	a:a	12 (fc:	aa:	14:	6a:9	9a:a2)							
Sen	Ide	rΙ	Pa	ddr	ess	: 1	92.	168	.1.1	00														
Tar	ge	tМ	AC	add	res	s:	Vmw	are_	_d1:	8e:	0c	(00):0c	:29	:d1	:8e:	:0c)							
Tar	.ue.	t T	Рa	ddr	P55	· 1	92	168	1.1	0.3														
		_	~ ~	14	_	_	~			_							,			_				
0000	90	ΘС	29	d1	8e	0C	ŤC	aa	14	6a	9a	a2	08	06	00	01)	· · ·]	• • •				
9010	98	00	06	04	00	02	fc	aa	14	6a	9a	a2	C0	a8	01	64] <u></u>	d				
9020	90	0c	29	d1	8e	0c	C0	a8	01	67	00	00	00	00	00	00	···) · · ·		g					
0030 G	90	00	00	00	00	00	00	00	00	00	00	00												

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 **reques**t 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
```

5 of 33

渿

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	Header	Protocol	Source IP	Destination IP
(20 bytes)	Brown	Red	Pink	Orango
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.ac	ldr =	= 192	2.168	.1.10	0 i	icmp																				
р.		Time		Sou	rce			Dest	inati	on			Proto	c L	ength	In	fo	_									
÷	4	2.62	289	192	.168	.1.	103	192	.16	8.1	.10	0	ICMF)	42	2 E	cho	(ping	3)	requ	les	t	id=	=0x7	′f84	, se
-	5	2.62	290	192	2.168	3.1.	100	192	.168	8.1	.10	3	ICMF)	60	9 E	cho	(ping	J)	repi	Ly		id=	=0x7	'f84	, se
	6	2.62	290	192	2.168	3.1.	103	192	.168	8.1	.10	0	ТСР		- 58	35	136	2	→ 44	13	[SYI	۱] ۱	Seq	=0	Wir	1=10	24 l
	- 7	2.62	291	192	2.168	3.1.	100	192	.168	8.1	.10	3	тср		60	94	43	-•	5136	62	[RS]	Γ, Ι	ACK] \$	Seq=	:1 A	ck=:
							0	TAY IT	າກປ	he	al.	an	ണ	£1	db	aí	in.										
T	Fra	me 4	: 42	2 byt	ces (on w	ire	(33	6 b	its	5),	42	byte	es	capt	ur	ed	(3	36 b	bit	s) (on	int	erf	face	9 0	
	Ethe	erne	t II	Ε, Ši	c: ۱	/mwa	re_	d1:8	e:0	с (00:	0c:	29:0	:11	8e:0	c)	, D	st	: Gi	iga	-Byt	<u>6</u>	a:9	a:a	a2 (fc:	aa::
	Inte	erne	t Pr	oto	col \	/ers	ion	4,	Src	: 1	92.	168	3.1.3	103	, Ds	t:	19	2.	168.	1.	100						
	Inte	erne	t Co	ontro	ol Me	essa	ige	Prot	oco	1																	
					-	-		-			-	-											-				
0	000	fc	aa	14 6	a 9a	a2	00	Θc	29	d1	8e	0C	08	00	45 (90			j)		Ε.				
0	910	00	1c	cd 4	5 00	00	38	01	31	80	C0	a8	01	67	C0 a	a8			Ε	8.	1	0	1				

020	01	64	08	00	78	7b	7f	84	00	00	.dx{	

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.

Inte	ternet 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]																
T	ype:	8	(Ec	ho	(pi	.ng)	re	que	st)								
C	ode:	Θ															
C	heck	sum	1: O	x78	7b	[co	rre	ect]									
[(Chec	ksu	Im S	tat	us:	Go	od]			2-0	~	8-					
I	dent	ifi	er	(BE):	326	44	(0x)	7f84	1)	92	illi.					
I	dent	ifi	er	(LE):	339	19	(0x8	847f)							
S	eque	nce	e nu	mbe	er (BE)	: 0) (0)	x000	00)							
S	eque	nce	e nu	mbe	er (LE)	: 0) (0)	x000	00)							
[]	Resp	ons	e f	ram	ie:	5]											
000	£		4.4	6.0	0.0	- 0	00	0.5	20	.	0.0	0.0	00	~~	45	00	
000	тс	aa	14	6a	9a	a2	00	⊎C	29	d1	8e	⊎C	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.

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

-

Inte	erne	t C	ont	rol	Ме	ssa	ge	Pro	toco	1							
T	Type: 0 (Echo (ping) reply) Code: 0 Chackaum: 0x807h [correct]																
C	ode:	0															
CI	heck	sum	n: 0	x80)7b	[co	rre	ect]									
ି [(Chec	ksu	Im S	stat	us:	Go	od]	1eT	ាា	<u>la</u>	સી	5					
I	dent	ifi	er	(BE):	326	44	(0x	7f84)							
I	dent	ifi	er	(LE	:(339	19	(0x	847f	•)							
Se	eque	ence	e nu	imbe	er (BE)	: 0) (0	x000	0))))							
Se	eque	ence	e nu	ımbe	er (LE)	: 0) (0	x000	00)							
5 pbc	Requ	est	fr	ame	: 4	1											
Ī	Resp	ons	e t	ime	: 0	.16	51 m	ıs]									
								-									
000	00	0c	29	d1	8e	0c	fc	aa	14	6a	9a	a2	08	00	45	00	
010	00	1c	66	с9	00	00	80	01	4f	fc	C0	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 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 m_{U_1}

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 ³² gart	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

						_		the second se	and the second second	_		_		_	_		
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	.,.>3rg
920	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 C (CS-III)	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,

	mmm bealthrandlabalta																		
000	00	0c	29	d1	8e	0c	fc	aa	14	6a	9a	a2	08	00	45	00)	 .jE.	
010	00	28	66	ca	40	00	80	06	0f	ea	c0	a8	01	64	сO	a8	.(f.@	 d	
020	01	67	91	bb	c8	a2	00	00	00	00	bc	af	75	69	50	14	.g	 uiP.	
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

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	Header	Protocol	Source IP	Destination IP
(20 bytes)	length	nakihana	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 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.

	ip.ac	ddr =	= 19	2.1	58.1	.100)																X	→	•	Expre	ssion		+
No		Time		S	ourc	e			0	Destin	atio	n		Prot	oc L	.engt	h Ir	fo											
	13 14 15 16	9.93 9.93 9.93 9.93	566. 568. 568. 571.	. 1 . 1 . 1	92. 92. 92. 92.	168 168 168 168	.1. .1. .1. .1.	103 100 103 103	W	192.1 192.1 192.1 192.1	168 168 168 168	.1. 110	1 1 1	TCP TCP TCP TCP	EG		74 3 56 8 54 3 54 3	4724 0 → 4724 4724	1 → 347 1 →	80 24 80 80	[SYN [SYN [ACK [RST] Se , A0] Se , A0	eq=0 CK] eq=1 CK]	Wir Seq= Ack Seq=	n=2 =0 k=1 =1	9200 Ack= Win Ack=	Ler 1 Wi =293 1 Wi	n=0 In=6 312 In=2	MSS 553 Len 931
• •	Fra Eth Int	me 1 erne erne	3: t I t P	74 I, rot	byt Src oco	es : V l V	on /mwa /ers	wir re_ ion	e (d1: 4,	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: .16	2 b Gig 8.1	its) a-By1 .100	on :_6a	int :9a	erfa :a2	ace (f	0 c:aa	:14:	6a:	• 9a:
Þ	Tra	nsmi	ssi	on	Con	tro	ol (P	rot	000	1, S	rc	Por	t:	347	24,	Ds	t P	ort:	80	, s	eq: (), L	.en:	Θ					
)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 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> c0 a0 f5	00 a8 02 0c	 . (. /	j. <.}@ d J.).@. .P)	g	E. 						4

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.

.d.b.PV. !W....`.

.

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-Q-Q-Q-	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
0000
     fc aa 14 6a 9a a2 00 0c
                              29 d1 8e 0c 08 00 45 00
                                                         ...j.... )....E.
     00 2c ea 8e 00 00 38 06
                              14 22 c0 a8 01 67 c0 a8
                                                         .,...8. ."...g..
010
```

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

21 57 00 00 00 00 60 02

Bits Color Light Brown Yellow Green TCP-SYN/ACK Packets Hex value 00 50 92 62 0x12 Decimal Value 80 28498 18	TCP Header	Source Port	Destination Port	Hex value of Flag
TCP-SYN/ACK Packets Hex value 00 50 92 62 0x12 Decimal Value 80 28498 18	Bits Color	Light Brown	Yellow	Green
Decimal Value 80 29499 19	TCP-SYN/ACK Packets Hex value	00 50 6 6 6 10 9	92 62 0 0 0 0	0x12
	Decimal Value	80	38498	18

05 b4

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".

)020

0030

01 64 96 62 00 50 56 0b

04 00 01 f6 00 00 02 04

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] Decol for a final sector of the secto
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)jE.
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.RV.!X`.
030 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 Coo fo	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.

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 <u>28 6a 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. 00 00 1d af 00 00 0030

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

<pre>root@kali:~# nmap -sT -p80 192.168.1.100</pre>
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	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.

*

Forensic Investigation of Nmap Scan using Wireshark -...

	ip.ac	ldr =	= 19	2.16	8.1.1	100																	X		-) E	xpres	sion	.	+
No		Time		So	urce				D)estin	atio	n		Prot	oc L	engt	h In	fo											
	13 14 15 16	9.95 9.95 9.95 9.95	566 568 568 571	19 19 19 19	2.1 2.1 2.1 <mark>2.1</mark>	68 68 68 68	.1. .1.: .1.:	103 100 103 103		92. 92. 92.	168 168 168 168			TCP TCP TCP TCP	ar		4 34 6 80 4 34 4 34	4724 9 + 4724 4724	, → 347 , →	80 24 80 80	[SYN] [SYN, [ACK] [RST,	Se AC Se AC	q=0 K] q=1 K]	Win Seq= Ack Seq=	=29 0 A =1 1 A	0200 Ack=1 Win= Ack=1	Len: Wir 293: Wir	=0 M n=65 12 L n=29	1SS 553 .en 931
•	Fra Eth	me 1 erne erne	3: 1 t II t Pi	74 k I, S	yte Src:	es Vi	on mwa ers	wir re_ ion	e (d1: 4,	592 8e:0 Src	bit c (s), 00: 92.	74 0c: 168	by 29:	tes d1: 103	ca 8e:0	ptur 0c), st:	red Ds 192	(59) t: .16	2 b Gig 8.1	its) a-Byt .100	on _6a	int :9a	erfa :a2	ce (fc	0 :aa:	14:6)a:9	a:
	Tra	nsmi	ssi	on C	ont	ro	1 P	rot	oco	1, S	rc	Por	t:	347	24,	Ds	t Po	ort:	80	, s	eq: 0	, L	en:	Θ					
	000 010 020 030 040	fc 00 01 72 fa	aa 3c 64 10 5e	14 ee 87 84 00	6a 9 7d 4 a4 (4a (90 (9a 40 00 00	a2 00 50 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> c0 a0 f5	00 a8 02 0c	 .d r.	j. <.}@ 1 .J.).@ <mark>.</mark> P)	E	I. 						•

• 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 WWW	Light Brown	Yellow es In	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] (relative sequence number) Sequence number: 0 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
 ...j...)...E.

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

 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 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 00 00 00 00	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]
€.
      00 0c 29 d1 8e 0c fc aa
                               14 6a 9a a2 08 00 45 00
                                                         ..)...E.
      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 fice

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 Coro 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 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



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

渿

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



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 Jengthom Co	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

ij	p.ad	dr =:	= 19	2.1	68.1	.104	ŀ																		×		•] E	Expression
р.	-	Time		S	ourc	e			Des	tinati	on		I	Prote	bc L	.ength	n Ir	nfo										
4	4	65.8	313.	. 1	92.	168	.1.	103	192	2.16	8.1	.10	4	тср		5	43	6956	j →	22	[F]	[N]	Se	q=1	Wi	.n=10	924	Len=0
4	1	65.9	914	. 1	92.	168	.1.	103	192	2.16	8.1	.10	4	TCP	2	5	43	6957	7 →	22	[F]	IN]	Se	q=1	Wi	.n=10	924	Len=0
					w	٨	<i>L</i> d	LC.	en a	щe	<u>u</u>	uus	110	201	IJ													
F	ram	ne 4	18:	54	by	tes	on	wi	re	(432	bi	ts)	, 5	4 b	yte	es ca	apt	ured	1 (4	432	bit	ts)	on	in	ter	face	e 0	
E	the	erne	tΙ	I,	Src	: v	mwa	re_	d1:	Be:0	с (00:	0c:	29:	d1:	8e:0	9c)	, Ds	st:	٧m	vare	e_6	b:7	1:a	7 (00:0	Dc:	29:6b:71:
I	nte	erne	tΡ	rot	осо	1 V	ers	ion	4,	Src	: 1	92.	168	.1.	103	3, Ds	st:	192	2.10	68.3	1.10	94						
Т	ran	ısmi	ssi	on	Con	tro	1 P	rot	oco.	1, S	rc	Por	tin	369	56,	Dst	:Р	ort:	22	2, 9	Seq	: 1	, L	en:	0			
					U	λĽ	U.	ШĊ	Lel 1	ալ	15	J.U.	gle	SH	U													
00	00	00	0c	29	6b	71	a7	00	0c	29	d1	8e	0c	08	00	45	00		.)k	q	.)		E	Ξ.				
01	10	00	28	6f	28	00	00	35	06	92	88	c0	a8	01	67	сø	a8		(0(5			.g.					
02	20	01	68	90	5c	00	16	60	a9	71	a7	00	00	00	00	50	01		h.\	· · `	. q		F	·.				
03	30	04	00	c5	00	00	00											•		••								

• Analysis TCP Header Details

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

Bits Color Light Brown Yellow Green TCP – FIN Packets Hex value 90 5c 00 16 01 Desimal Value 26056 22 1	TCP Header	Source Port	Destination Port	Hex value of Flag
TCP-FIN Packets Hex value 90 5c 00 16 01 Desimal Value 36056 32 1	Bits Color	Light Brown	Yellow	Green
Desimal Value 26056 22	TCP-FIN Packets Hex value	90 5c	0016	01 falaafa
Decimal value 30550 22 0 1	Decimal Value	36956	22	hararann

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

渿

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	Header length	Protocol	Source IP	Destination IP
(20 bytes) 🛛 🕦	ww.hack	ngartie	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 =	= 19	92.1	68.1	.104	ļ																		X		-	Expr	ession
	Time		S	ourc	e			Des	tinati	on		1	Proto	oc L	engtl	h Ir	nfo											
7	3.3	887. 892.	. 1	92 92	$168 \\ 168$	40	103 103	192 192	2.16	8.1 8.1	.10	4	TCP TCP	0	5 5	44	491 491	.8 - .9 -	+ 22 + 22	2 [2 [<no <no< td=""><td>ne> ne></td><td>] 5</td><td>Seq= Seq=</td><td>1 \ 1 \</td><td>Win= Win=</td><td>1024</td><td>Len: Len:</td></no<></no 	ne> ne>] 5	Seq= Seq=	1 \ 1 \	Win= Win=	1024	Len: Len:
Era	mo 7	. 5	4 h	vto	c 0	n	iro	(1)	22 h	ite	1	54	byt	00	can	+	od	(13	22 1	,it	c)	00.	int	orf	200	0.0		Þ
Eth	erne	t I	Ξ,	Src	: V	mwa	re_	d1:	32 D Be:0	c (00:	0c:	29:	d1:	8e:	0c)	, D	(4) st	: Vn	nwa	s) re_	6b:	71:	a7	(0)	0:0c	:29:	6b:71
Int Tra	erne nsmi	t P .ssi	rot .on	oco Con	l V tro	ers l P	ion rot	4, 000	Src	: 1 rc	.92. Por	168 t:	449	103	, D	st: t P	19 ort	2.1	168. 22.	.1. Se	104 a:	1. I	Ler	n: 0				
					u		uuc		GN	ų	1.1	iar	919	сл.														
000	00	0c	29	6b	71	a7	00	0c	29	d1	8e	0c	08	00	45	00)	kq.).		Ε.					
910	00	28	e9	26	00	00	31 h1	06	10	8a	C0	a8	01	67	C0	a8		. (.	&	1.	•••	g	 D					
930	04	00	df	70 31	00	00	DT	04	е1	01	00	00	00	00	50	00			v 1	•••	•••		г.					

• 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	Af76	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 TC+-

≈

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]
                        (relative sequence number)
  Sequence number: 1
  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
920	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.

<pre>root@kali:~# nmap -s></pre>	(-p22 192.168.1.104
Starting Nmap 7.60 (Nmap scan report for Host is up (0.00020s	https://nmap.org) at 2018-01-09 08:43 EST 192.168.1.104 latency).
PORT STATE	SERVICE
22/tcp <mark>open</mark> filtered	ssh
MAC Address: 00:0C:29	9:6B:71:A7 (VMware)
Nmap done: 1 IP addre	ess (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	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.ac	ddr ==	= 19	2.1	68.1	.104	ł																		×∣→
).	Time		S	ourc	e			Dest	tinati	ion		1	Prote	oc Le	ength	n In	fo							
9 10	2.78	3 <mark>62</mark> 371	. 1	92. 92.	168 168	.1. .1.	103 103	192 192	.16	8.1 8.1	.10	4 4	TCP TCP		5 5	45 45	2469 2470) → →	22 22	[FI	N, N,	PSH, PSH,	U	RG] S
							Ŵ	NW	<u>Hit</u>	<u>i</u>	КП	Q	Jit	C	es	Tr								
Fra Eth Int <mark>Tra</mark>	me 9 erne erne <mark>nsmi</mark> :	: 5 t I t P ssi	4 b I, rot <mark>on</mark>	yte Src oco <mark>Con</mark>	s o : V l V <mark>tro</mark>	n w mwa ers l P	/ire_ ire_ ion <mark>Prot</mark>	(43 d1:8 4, <mark>oco</mark>]	32 b 3e:0 Src ., S	its c (: 1 crc), 00: 92. Por	54 0c: 168 t:	byt 29: .1. <mark>524</mark>	es d1: 103 <mark>69,</mark>	capt 8e:0 , Ds Dst	tur 0c) st: t P	ed (, Ds 192 ort:	432 t: .16 22	2 bi Vmv 88.1	its) vare_ L.104 Seq:	on _6b 4 1,	n int 0:71: Len	er a7 :	face (00 0
000 010 020 030	00 00 01 04	0c 28 68 00	29 b5 cc f3	6b 7e f5 82	71 00 00 00	a7 00 16 00	00 34 78	0c 06 66	29 4d ee	d1 32 a6	8e c0 00	0c a8 00	08 01 00	00 67 00	4 <u>5</u> c0 50	00 a8 29	. .	.)k (.~ 1	q 4 x	.). . M2 f	 	E. .g P)		

• 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

*

.



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.



Step to Identify NMAP UDP Scan

Collect IP Header Details for Protocol Version

渿

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

```
      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	Θc	29	d1	8e	0c	08	00	45	00)kq)E.
910	00	1c	15	d3	00	00	2c	11	f4	de	сO	a8	01	67	c0	a8	, .	g
020	01	68	82	75	00	44	00	08	f9	04							.h.u.D	

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.

User	⁻ Da	tag	ram	Pr	oto	col	, Sr	c Port	: 3	3339	7,	Dst	Po	rt:	68	
S	Source Port: 33397															
De	Destination Port: 68															
Le	engt	h:	8	777	A' 'A			ાણન	JU	ાલાર	151	JU				
Cl	neck	sun	1: C	xf9	04	[un	veri	fied]								
[(Chec	ksι	ım S	stat	us:	Un	veri	fied]								
[\$	Stre	am	ind	lex:	1]											
0000	00	0c	29	6b	71	a7	00 0	c 29	d1	8e	0c	08	00	45	00)kq)E.
010	~ ~	4 -	1 5	42	00	00	2c 1	1 f4	de	C0	a8	01	67	C0	a8	n
DIO	00	TC	12	us	00	00	20 3	± 14								·· <u>···</u> ···y··
020	00 01	1C 68	82	75	<u>90</u>	44	00 0	8 f9	04						uo	.h <mark>.u.D</mark>

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



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



Srinivas August 8, 2019 at 10:14 am

Nice Article

*



<u>David Mata</u>

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





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

≈

_

• -

• • -

-

_

—

- -

*