

Detect/Analyze Scanning Traffic Using Wireshark

"Wireshark", the world's most popular Network Protocol Analyzer is a multipurpose tool. It can be used as a Packet Sniffer, Network Analyser, Protocol Analyser & Forensic tool. Through this article my focus is on how to use Wireshark to detect/analyze any scanning & suspect traffic.

et's start with Scanning first. As a thief studies surroundings before stealing something from a target, similarly attackers or hackers also perform foot printing and scanning before the

actual attack. In this phase, they want to collect all possible information about the target so that they can plan their attack accordingly. If we talk about scanning here they want to collect details like:

icmp_filtered.pcapng [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10]	0		
Eile Edit View Go Capture Analyze Statistics Telephony Tools	Internals Help		
004 = 4 = 5 2 2 4 4 4 5 5	E QQQ 🖸 📓 🖉 🥵 %	1 1	
Filter is the -R or is the time -0	Expression Clear Apply Store		
Filter: Icmp.type==0 or icmp.type==0	Expression Clear Apply Save		
No. Time Source Destination 0.00804200192.168.145.1 192.168.145.12 192.168.145.12 5.00946100192.168.145.1 192.168.145.12 192.168.145.12 6.0.0094800192.168.145.1 192.168.145.13 192.168.145.13 7.0.03988200192.168.145.1 192.168.145.13 192.168.145.13 9.0.04076700192.168.145.1 192.168.145.13 192.168.145.13 10.0.4099000192.168.145.13 192.168.145.13 192.168.145.13 10.0.4099000192.168.145.131 192.168.145.131 192.168.145.131 10.0.40170800192.168.145.131 192.168.145.132 192.168.145.132 14.0.06017200192.168.145.132 192.168.145.132 192.168.145.132 15.0.06096700192.168.145.132 192.168.145.132 192.168.145.132 16.0.06096700192.168.145.132 192.168.145.132 192.168.145.132 17.0.06155700192.168.145.132 192.168.145.132 192.168.145.132 19.0.07974700192.168.145.132 192.168.145.132 192.168.145.132 10.0.0802800192.168.145.133 192.168.145.133 192.168.145.133 20.08076000192.168.145.133 192.168.145.133 192.168.145.133	Protocol Length Into ICMP % Echo (ping) request ICMP % Echo (ping) reply ICMP % Echo (ping) reply ICMP % Echo (ping) reply ICMP % Echo (ping) reply ICMP % Echo (ping) request ICMP % Echo (ping) request ICMP % Echo (ping) request ICMP % Echo (ping) request ICMP % Echo (ping) reply ICMP % Echo (ping) request ICMP % Echo (ping) request	<pre>1d=0x0001, seq=332/19457, ttl=255 (reply in 4) id=0x0001, seq=332/19457, ttl=228 (request in 3) id=0x0001, seq=333/19713, ttl=255 (reply in 6) id=0x0001, seq=335/2025, ttl=64 (request in 5) id=0x0001, seq=336/20481, ttl=255 (reply in 10) id=0x0001, seq=336/20481, ttl=255 (reply in 110) id=0x0001, seq=336/20481, ttl=255 (reply in 12) id=0x0001, seq=336/20481, ttl=255 (reply in 12) id=0x0001, seq=338/20993, ttl=255 (reply in 12) id=0x0001, seq=338/20993, ttl=255 (reply in 12) id=0x0001, seq=338/20993, ttl=228 (request in 13) id=0x0001, seq=338/21249, ttl=255 (reply in 14) id=0x0001, seq=339/21249, ttl=255 (reply in 16) id=0x0001, seq=342/21249, ttl=255 (reply in 17) id=0x0001, seq=342/21261, ttl=255 (reply in 20) id=0x0001, seq=341/21761, ttl=255 (reply in 22) id=0x0001, seq=342/22017, ttl=255 (reply in 22) id=0x0001, seq=342/22017, ttl=255 (reply in 22) id=0x0001, seq=342/22273, ttl=255 (reply in 23) id=0x0001, seq=342/22273, ttl=255 (request in 23) id=0x0001, seq=342/22273, ttl=255 (request in 23) id=0x0001, seq=343/22273, ttl=255 (reply in 24) id=0x0001, seq=343/2273, ttl=255 (reply in 24) id=0x0001, seq=343/22</pre>	
30 9.30293800 192.168.145.1 192.168.145.254	ICMP 98 Echo (ping) request	id=0x0001, seq=762/64002, tt1=255	•
<pre>Interact control Massing - Potocol Type: 8 (Echo (ping) request Code: 0 Checkstm: 04501 [contrect] Identifier (BE): 1 (0x0001) Identifier (LE): 256 (0x0100)</pre>			- -
0020 01 fe 08 00 f5 04 00 01 02 f1 00 0	00 · · · · · · · · · · · · · · · · · ·		* 11. 1
Internet Control Message Protocol (icmp), 6 Packets: 30 · Displayed	27 (90.0%) · Load time: 0:00.001	Profile: Default	

Figure 1. Ping Sweep

- Which IP addresses are in use?
- · Which port/services are active on those IPs?
- · Which platform (Operating System) is in use?
- What are the vulnerabilities & other similar kinds of information.
- Now I am moving to some popular scan methods and how to detect them in Wireshark.

Ping Sweep

This scan is helpful to find out which IPs are active in the network. Ping Sweep can be performed using ICMP, TCP or UDP, the most popular one is ICMP Ping Sweep. In this ICMP type 8, ECHO request is followed by ICMP type 0, ECHO reply packets are being used while in TCP/UDP ping sweep packets are destined to TCP/UDP port 7, The ECHO port. If that target host doesn't support ECHO service then this TCP/UDP ping sweep will not work. Thus ICMP ping sweep is mostly used, but if there is a firewall in between which is configured to block ICMP packet then even ICMP ping sweep is useless. In this situation, ARP scan/ARP sweep can be used which is discussed next (Figure 1).

To detect ICMP ping sweep in Wireshark apply simple filter icmp.type==8 or icmp.type==0. TCP
ping sweep can be detected with tcp.dstport==7
filter and for UDP ping sweep udp.dstport==7 filter
can be used. After applying these filters if we are
getting more than expected packets then it's possible that ping sweep is going on in our network. We need to be careful about the volume of such traffic as it might be normal ping traffic. It should be considered as a scan signature only if you are getting unexpected increase in ICMP traffic.

ARP Sweep/ARP Scan

As discussed in previous scan that if a firewall is implemented in between and ICMP is blocked then we can't use ICMP ping sweep. In such a situation, ARP scan is helpful to find out active IPs in the network. Here, attacker sends ARP broadcast (for broadcast, destination MAC will be 0xff:ff:ff:ff:ff:ff:ff for each and every possible IP in selected subnet and if he gets ARP response then it shows that IP is active. Advantage of this scan is that ARP communication can't be filtered or disabled because all TCP/IP communication is based on it. Blocking or disabling ARP communication will break TCP/IP communication or it will force static ARP entries and disadvantage of this scan is that it can't cross layer 3 Devices. This scan can be easily detected with filter ARP. After applying this filter if we are getting unexpected no. of ARP queries as shown in the picture, it is a sign for ARP scan or ARP sweep (Figure 2).

TCP Half Open/Stealth Scan

To detect open or close TCP port on target system, Stealth scan is the most often used method. In this scan, attacker sends a SYN packet on the target

Ping_Sweep_Scan.pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.	10]]	
Eile Edit View Go Capture Analyze Statistics Telephony Iools	Internals Help	
0 0 1 = 1 = 2 2 2 4 4 4 4 5 2	🗐 🕞 Q, Q, Q, 🖸 👹 🖾 🥵 🔆 🙀	
Filter	Forression Clear Annly Save	
No. Time Sever Destination	Destand Leasth Infe	
No. Time Source Destination	Protocol Length Into	
13 0.616678 Vmware_a5:0d:e8 Broadcast	ARP 60 who has 192.168.10.37 Tell 192.168.10.1	
15.0.617027 Vmware a5:0d:e8 Broadcast	ARP 60 who has 122 168 10.57 Tell 192 168 10.1	
16 0.617030 Vmware_a5:0d:e8 Broadcast	ARP 60 who has 192.168.10.6? Tell 192.168.10.1 Attacker IP	
17 0.619412 Vmware_a5:0d:e8 Broadcast	ARP 60 who has 192.168.10.7? Tell 192.168.10.1 192.168.10.1	
18 0.619782 Vmware_a5:0d:e8 Broadcast	ARP 60 who has 192.168.10.8? Tell 192.168.10.1	
19 0.619786 vmware_a5:0d:e8 Broadcast	ARP 60 who has 192.168.10.97 Tell 192.168.10.1	
20 0.619789 Vmware_a5:0d:e8 Broadcast	ARP 60 Who has 192.168.10.10? Tel 192.168.10.1	
21 0.019/91 Vinware_a5:0d:e8 Broadcast	ARP 60 Who has 192.105.10.11; 1011 192.105.10.1	
capture center, or pytes that pits	AP 00 MIN Has 132.100.10.12: 101 192.100.10.1	
[Frame is marked: False]		^
[Frame is ignored: False]		
[Protocols in frame: eth:arp]		
[Coloring Rule Name: ARP]		
[Coloring Rule String: arp]	(AR) Art: Broadcast (144.64.64.64.64.64)	
Destination: Broadcast (ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:	(es), bsc. broadcast ((1.11.11.11.11))	
<pre>B Source: vmware_a5:0d:e8 (00:0c:29:a5:0d:e8)</pre>		
Type: ARP (0x0806)		
Padding: 000000000000000000000000000000000000		
Address Resolution Protocol (request)		
Hardware type: Ethernet (1)		
Protocol type: IP (0x0800)		=
Protocol size: 4		
opcode: request (1)		
Sender MAC address: Vmware_a5:0d:e8 (00:0c:29:a)	::0d:e8)	
Sender IP address: 192.168.10.1 (192.168.10.1)		
Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff		
Target IP address: 192.168.10.0 (192.168.10.0)		*
0000 ff ff ff ff ff ff <u>00 0c 29 a5 0d e8</u> 08 06 00	01	
0010 08 00 06 04 00 01 00 0c 29 a5 0d e8 c0 a8 0a	01)	
Source Hardware Address (eth.src), 6 bytes Packets: 510 - Displaye	d: 510 (100.0%) - Load time: 0:00.016 Prof.	le: Default

Figure 2. ARP Scan/ARP sweep



port like a normal TCP communication. If the port is open, he will get SYN+ACK and RST or RST+ACK if the port is closed. After getting SYN+ACK on the open port as a response, attacker will send RST because attacker doesn't want to open TCP session with a target. If that target port is firewalled then expected response is ICMP type 3 Packet with Code 1,2,3,9,10, or 13. So in Wireshark if we are getting a lot of RST packets or ICMP type 3 packets, it can be a sign for Stealth Scan or TCP Full Connect Scan. As we can see that in the above picture a lot of SYN & RST packets are moving back and forth, but there is no data communication between these hosts. To get a quick view in the above capture we can go to top menu Statistics -> Conversations and then go to TCP tab. There we can see multiple TCP sessions, but all are having less than 4 packet communications which is a sign for TCP port Scan (Figure 4).

TCP Full Connect Scan

In this scan attacker is going to perform complete three way hand shake to find out if the port is open

Stea	Ith Scan.pcap [W	Vireshark 1.10.2 (SVI	N Rev 51934 from /trunk-1.10)]			
Eile f	Edit View Go	Capture Analyze	Statistics Telephony Tools	Internals Help		
0 6	A = A	8 8 8 2	Q & & @ 7 2		ର୍ ପ୍ 🗹 👹	M 🥵 🔆 🙀
Filter:				Expression	Clear Apply Save	
No.	Time	Source	Destination	Protocol L	ength Info	*
	37 13.014532	192.168.10.1	192.168.10.2	TCP	60 54433 > rt	sp [SYN] Seq=0 Wir=1024 Len=0 MSS=1460
	38 13.014549	192.168.10.2	192.168.10.1	TCP	54 telnet > 1	1433 [RST, ACK] Seq=1 Ack=1 win=0 Len=0
	39 13.014591	192.168.10.2	192.168.10.1	TCP	54 imaps > 54	433 [RST, ACK] Sec-1 Ack-1 win-0 Len-0
4	40 13.014644	192.168.10.2	192.168.10.1	TCP	54 rtsp > 54	B3 [RST, ACK] Seq=1 Ack=1 win=0 Len=0
4	41 13.014698	192.168.10.1	192.168.10.2	TCP	60 54433 > m	icrosoft-ds [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	42 13.014757	192.168.10.2	192.168.10.1	TCP	58 microsoft	ds > 54433 [SYN, /CK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
4	43 13.014836	192.168.10.1	192.168.10.2	TCP	60 54433 > i	ent [SYN] Seq=0 Win=1024 Len=0 M55=1460
4	44 13.014852	192.168.10.2	192.168.10.1	тср	54 ident > 54	433 [RST, ACK] Sed=1 Ack=1 win=0 Len=0
4	45 13.014914	192.168.10.1	192.168.10.2	TCP	60 54433 > m	crosoft-ds [RST] Seq=1 Win=0 Len=0
4	46 13.017271	192.168.10.1	192.168.10.2	TCP	60 54433 > ht	tp-alt [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	47 13.017316	192.168.10.2	192.168.10.1	TCP	54 http-alt :	54433 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	48 13.017438	192.168.10.1	192.168.10.2	TCP	60 54433 > pj	tp [SYN] Seq=0 wir=1024 Len=0 MSS=1460
4	49 13.017440	192.168.10.1	192.168.10.2	TCP	60 54433 > s:	sh [SYN] Seq=0 Win=1024 Len=0 MSS=1460
	50 13.017441	192.168.10.1	192.168.10.2	TCP	60 54433 > n	tbios-ssn [SYN] Seq=0 Win=1024 Len=0 MSS=1460
	51 13.017483	192.168.10.2	192.168.10.1	TCP	54 pptp > 54	83 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
	52 13.017531	192.168.10.2	192.168.10.1	TCP	54 ssh > 544	3 [RST, ACK] Seq=1 ACK=1 win=0 Len=0
	53 13.017623	192.168.10.2	192.168.10.1	TCP	58 netbios-s:	n > 54433 [SYN, ACK] Seq=0 Ack=1 W1n=8192 Len=0 M55=1460
10	54 13.017685	192.168.10.1	192.168.10.2	TCP	60 54433 > ra	id-am [SYN] Seq=0 win=1024 Len=0 MSS=1460
	55 13.017687	192.168.10.1	192.168.10.2	TCP	60 54433 > di	idi [SYN] Seq=0 Win=1024 Len=0 MSS=1460
	56 13.017687	192.168.10.1	192.168.10.2	TCP	60 54433 > 1	p [SYN] Seq=0 win=1024 Len=0 MSS=1460
	57 13.017704	192.168.10.2	192.168.10.1	TCP	54 raid-am >	54433 [RST, ACK] Seq=1 Ack=1 win=0 Len=0
	58 13.017800	192.168.10.2	192.168.10.1	TCP	54 dmidi > 54	433 [RST, ACK] Sed=1 Ack=1 Win=0 Len=0
	59 13.017866	192.168.10.2	192.168.10.1	TCP	54 ipp > 544	3 [RST, ACK] Seq-1 Ack-1 Win-0 Len-0
	60 13.017938	192.168.10.1	192.168.10.2	TCP	60 54433 > ni	tbios-ssn [RST] Seq=1 Win=0 Len=0
	61 13.017940	192.168.10.1	192.168.10.2	TCP	60 54433 > ci	mmplex_link [syn] seq=0 win=1024 Len=0 Mss=1460
1	62 13.017941	192.168.10.1	192.168.10.2	TCP	60 54433 > di	x [SYN] Seq=0 Win=1024 Len=0 MSS=1460
	63 13.017986	192.168.10.2	192.168.10.1	TCP	54 commplex-	ink > 54433 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
	64 13.018077	192.168.10.2	192.168.10.1	TCP	54 dnx > 544	3 [RST, ACK] Seq=1 ACK=1 Win=0 Len=0

Figure 3. Stealth Scan

Conv	ersations: Stealth S	ican.pcap									
1455 Etherr	et: 3 Fibre Chann	CI FDDI IP	4:2 IPv6 IPX JX	TA NCP RSVP SCTP TCP:	101 Token	Ring UDP:1 USB	WLAN				
145					TCP Con	versations					
464 Hours	SS A	PORA	Address B	POIL B PACKELS	bytes	Packets A-	B 4 Bytes A-B	Packets A	-B + Bytes A+-B	 Rel St 	tart 🔺
47: 192.1	68.10.1	54433	192,168,10,2	sunroc	2	114	1	60	1	54	13,005696(
48 192.1	68.10.1	54433	192.168.10.2	smux	2	114	1	60	1	54	13.0085240
48 192.1	68.10.1	54433	192.168.10.2	ddi-tcp-1	2	114	1	60	1	54	13.008526
491 192.1	68.10.1	54433	192.168.10.2	rap	2	114	1	60	1	54	13.011190
727 192.1	68.10.1	54433	192.168.10.2	http	2	114	1	60	1	54	13.011321(
74 192.1	68.10.1	54433	192.168.10.2	submissic	2	114	1	60	1	54	13.011323
744 192.1	68.10.1	54433	192.168.10.2	epmap	3	178	2	120	1	58	13.011459
744 192.1	68.10.1	54433	192.168.10.2	blackjack	2	114	1	60	1	54	13.0142000
748 192.1	68.10.1	54433	192.168.10.2	domain	2	114	1	60	1	54	13.014328
75 192.1	68.10.1	54433	192.168.10.2	ftp	2	114	1	60	1	54	13.014348
76/ 192.1	68.10.1	54433	192.168.10.2	telnet	2	114	1	60	1	54	13.014530
768 192.1	68.10.1	54433	192.168.10.2	imaps	2	114	1	60	1	54	13.014531(
768 192.1	68.10.1	54433	192.168.10.2	rtsp	2	114	1	60	1	54	13.014532
770 192.1	68.10.1	54433	192.168.10.2	microsoft	3	178	2	120	1	58	13.014698/
780 192.1	68.10.1	54433	192.168.10.2	ident	2	114	1	60	1	54	13.014836
78(192.1	68.10.1	54433	192.168.10.2	http-alt	2	114	1	60	1	54	13.0172710
.79 192.1	68.10.1	54433	192.168.10.2	pptp	2	114	1	60	1	54	13.017438
794 192.1	68.10.1	54433	192.168.10.2	ssh	2	114	1	60	1	54	13.0174400
798	68.10.1	54433	192.168.10.2	netbios-s	3	178	2	120	1	58	13.017441(
807	68.10.1	54433	192.168.10.2	raid-am	2	114	1	60	1	54	13.017685(-
81											•
O E V Nar	me resolution	Limit to disp	olay filter								
÷,											

Figure 4. Statistics->Conversation_ TCP tab



or close. Attacker will send SYN on target port, if the port opens, he will get SYN+ACK and RST+ACK on the closed port. After getting SYN+ACK, the attacker will send ACK and try to establish TCP session and then terminate it. In Wireshark, we can use a similar method like TCP Half open scan to detect TCP full connect as well. If that target port is firewalled then here also we will get the same response which is ICMP type 3 Packet with Code 1,2,3,9,10, or 13. Following filters can be used in Wireshark to detect TCP scan packet quickly (TCP Half open & TCP Full Connect) To get SYN, SYN+ACK, RST & RST+ACK packet

tcp.flags==0x002 or tcp.flags==0x012 or tcp.
flags==0x004 or tcp.flags==0x014

• To get ICMP type 3 Packet with Code 1,2,3,9,10, or 13 Packet

icmp.type==3 and (icmp.code==1 or icmp.code==2 or icmp.code==3 or icmp.code==9 or icmp.code==10 or icmp.code==13)

Nul	Windows.pcap	[Wireshark 1.10.2 (S	VN Rev 51934 from /trunk-1.10)	1			
Eile	Edit <u>View</u> <u>Go</u>	Capture Analyze	Statistics Telephony Tools	Internals Hel	p		
0 6			19.00072			🔐 🗹 🥵 % 🛱	
Filter:	tcp.flags==0x0	00		 Expression. 	Clear Apply	Save	
No.	Time	Source	Destination	Protocol	Length Info		
	3 0.001150	192.168.10.1	192.168.10.2	TCP	60 56018 :	> sunrpc [<none>] Seq=1 win=1024 Len=0</none>	-
	4 0.001168	192.168.10.1	192.168.10.2	TCP	60 56018 :	<pre>> smtp [<none>] :eq=1 win=1024 Len=0</none></pre>	
	5 0.001169	192.168.10.1	192.168.10.2	TCP	60 56018 :	> submission [<none>] Seq=1 Win=1024 Len=0</none>	
	9 0.001442	192.168.10.1	192.168.10.2	TCP	60 56018 :	> mysql [<none>] Seq=1 Win=1024 Len=0</none>	
1	10 0.001443	192.168.10.1	192.168.10.2	TCP	60 56018 :	> imaps [<none>] Seq=1 Win=1024 Len=0</none>	
	11 0.001444	192.168.10.1	192.168.10.2	TCP	60 56018 :	> ftp [<none>] Sdq=1 win=1024 Len=0</none>	
	12 0.001444	192.168.10.1	192.168.10.2	TCP	60 56018 :	> pop3s [<none>] Seq=1 Win=1024 Len=0</none>	
	13 0.001445	192.168.10.1	192.168.10.2	TCP	60 56018 :	<pre>> http [<none>] Seq=1 Win=1024 Len=0</none></pre>	
3	14 0.001446	192.168.10.1	192.168.10.2	TCP	60 56018 :	> ddi-tcp-1 [<nore>] Seq=1 Win=1024 Len=0</nore>	
	15 0.001446	192.168.10.1	192.168.10.2	TCP	60 56018 :	<pre>> http-alt [<nond>] Seq=1 win=1024 Len=0</nond></pre>	
	23 0.001798	192.168.10.1	192.168.10.2	TCP	60 56018 :	> rap [<none>] Seq=1 Win=1024 Len=0</none>	
	24 0.001799	192.168.10.1	192.168.10.2	TCP	60 56018 :	> microsoft-ds [None>] Seq=1 Win=1024 Len=0	
	27 0.001982	192.168.10.1	192.168.10.2	TCP	60 56018 :	> pptp [<none>] seq=1 win=1024 Len=0</none>	
	28 0.001998	192.168.10.1	192.168.10.2	TCP	60 56018 :	> blackjack [<nore>] Seq=1 win=1024 Len=0</nore>	
	29 0.001998	192.168.10.1	192.168.10.2	TCP	60 56018 :	> ssh [<none>] Seq=1 Win=1024 Len=0</none>	
	30 0.001999	192.168.10.1	192.168.10.2	TCP	60 56018 :	> ident [<none>] Seq=1 Win=1024 Len=0</none>	
	31 0.002000	192.168.10.1	192.168.10.2	TCP	60 56018 :	> telnet [<none>] Seq=1 Win=1024 Len=0</none>	
	37 0.002244	192.168.10.1	192.168.10.2	TCP	60 56018 :	> ms-wbt-server <pre>seq=1 win=1024 Len=0</pre>	
	38 0.002259	192.168.10.1	192.168.10.2	TCP	60 56018 :	> imap [<none>] seq=1 win=1024 Len=0</none>	
	39 0.002260	192.168.10.1	192.168.10.2	TCP	60 56018 :	> rtsp [<none>] Seg=1 Win=1024 Len=0</none>	
	40 0.002261	192.168.10.1	192.168.10.2	TCP	60 56018 :	> pop3 [<none>] Seq=1 Win=1024 Len=0</none>	
	41 0.002262	192.168.10.1	192.168.10.2	TCP	60 56018 :	> h323hostcall [-None>] Seg=1 Win=1024 Len=0	
	42 0.002262	192.168.10.1	192.168.10.2	TCP	60 56018 :	> https [<none>] Seg=1 win=1024 Len=0</none>	
	43 0.002263	192.168.10.1	192.168.10.2	TCP	60 56018 :	> epmap [<none>] Seq=1 Win=1024 Len=0</none>	-
	Stroom inde	· · · ·					
	sequence nur	mber: 1 (re]	ative sequence number)			
	loader lenn	tr 20 hates	active sequence number	·			
10.0	lags: 0x00	(<none>)</none>					
-	indow size	value: 100					E
	Calculated	window size: 1	0241				
	Window size	e scaling facto	r: -1 (unknown)]				-
	111100W 3121	e scarmy races					

Figure 5. TCP Null Scan

	(mas_)	Windows.pcap	Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10	0]			- 0 - X
Eile	Edi	t View Go	Capture Analyze	Statistics Telephony Tools	Internals Help			
0	۲	A = <u>A</u>	🖻 🖨 🗶 🌮	। 🔍 🗢 🗢 🖓 🔮		Q Q 🗹 🛛	🎬 🗹 🥵 🖗 🔛	
Filt	er: to	p.flags==0x02	9		Expression	. Clear Apply	Save	
No.		Time	Source	Destination	Protocol	Length Info		*
	12	0.001679	192.168.10.1	192.168.10.2	TCP	60 46942	telnet [FIN, PSH, URG] Seg=1 Win=1024 Urg=0 Len=0	
	13	0.001680	192.168.10.1	192.168.10.2	TCP	60 46942	smux [FIN, PSH, URG] seg=1 v1n=1024 Urg=0 Len=0	
	14	0.001681	192.168.10.1	192.168.10.2	TCP	60 46942	epmap [FIN, PSH, URG] Seq=1 win=1024 Urg=0 Len=0	
	18	0.004959	192.168.10.1	192.168.10.2	TCP	60 46942 :	ssh [FIN, PSH, URG] Seq=1 W n=1024 Urg=0 Len=0	1
	19	0.004962	192.168.10.1	192.168.10.2	TCP	60 46942	pop3 [FIN, PSH, URG] Seg=1 vin=1024 Urg=0 Len=0	
	22	0.007623	192.168.10.1	192.168.10.2	TCP	60 46942 :	sunrpc [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0	
	24	0.007862	192.168.10.1	192.168.10.2	TCP	60 46942	blackjack [FIN, PSH, URG] sig=1 win=1024 Urg=0 Len=0	
	25	0.007865	192.168.10.1	192.168.10.2	TCP	60 46942	submission [FIN, PSH, URG] (eq=1 Win=1024 Urg=0 Len=0	
	26	0.007867	192.168.10.1	192.168.10.2	TCP	60 46942 :	ddi-tcp-1 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0	
	30	0.010765	192.168.10.1	192.168.10.2	TCP	60 46942	smtp [FIN, PSH, URG] seg=1 vin=1024 Urg=0 Len=0	1
	31	0.010768	192.168.10.1	192.168.10.2	TCP	60 46942	h323hostcall [FIN, PSH, URG] seq=1 win=1024 Urg=0 Len=0	
	32	0.010770	192.168.10.1	192.168.10.2	TCP	60 46942	ident [FIN, PSH, URG] seq=1 win=1024 Urg=0 Len=0	
	36	0.011204	192.168.10.1	192.168.10.2	TCP	60 46942 :	rtsp [FIN, PSH, URG] Seq=1 vin=1024 Urg=0 Len=0	
	37	0.011207	192.168.10.1	192.168.10.2	TCP	60 46942	microsoft-ds [FIN, PSH, URG Seq=1 Win=1024 Urg=0 Len=0	
	38	0.011209	192.168.10.1	192.168.10.2	TCP	60 46942	mysql [FIN, PSH, URG] seq=1 win=1024 Urg=0 Len=0	
	39	0.011211	192.168.10.1	192.168.10.2	TCP	60 46942	ms-wbt-server [FIN, PSH, URG] seq=1 win=1024 urg=0 Len=0	
	40	0.011213	192.168.10.1	192.168.10.2	TCP	60 46942	https [FIN, PSH, URG] Seq=1 win=1024 Urg=0 Len=0	
	46	0.014066	192.168.10.1	192.168.10.2	TCP	60 46942 :	pptp [FIN, PSH, URG] Seq=1 vin=1024 Urg=0 Len=0	
	47	0.014069	192.168.10.1	192.168.10.2	TCP	60 46942	rap [FIN, PSH, URG] seq=1 W n=1024 Urg=0 Len=0	
	50	0.014402	192.168.10.1	192.168.10.2	TCP	60 46942	rfb [FIN, PSH, URG] seq=1 win=1024 Urg=0 Len=0	
	51	0.014470	192.168.10.1	192.168.10.2	TCP	60 46942	domain [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0	
	52	0.014478	192.168.10.1	192.168.10.2	TCP	60 46942 :	50001 [FIN, PSH, URG] Seq=1 win=1024 Urg=0 Len=0	
	53	0.014480	192.168.10.1	192.168.10.2	TCP	60 46942	rsync [FIN, PSH, URG] seq=1 Win=1024 Urg=0 Len=0	
	58	0 014972	197 168 10 1	192 168 10 2	TCP	60 46942	9110 FEIN PSH URCI Senal Vinal024 URGa0 Lena0	
	[St	ream inde	x: 4]					
	Sec	quence num	ber: 1 (rel	ative sequence number))			
	He	der lengt	n: 20 bytes					
0	Fla	igs: 0x029	(FIN, PSH, UR	G)				-
	with	dow size	value: 1024					1
	[Ca	alculated	window size: 1	024]				
	[Wi	indow size	scalino facto	r: -1 (unknown)]				۲
гI	gu	reo./(_r xmas Sc	an				



To get SYN, SYN+ACK, RST & RST+ACK packet along with ICMP type 3 Packet with Code 1,2,3,9,10, or 13 Packet

tcp.flags==0x002 or tcp.flags==0x012 or tcp. flags==0x004 or tcp.flags==0x014 or (icmp.type==3 and (icmp.code==1 or icmp.code==2 or icmp. code==3 or icmp.code==9 or icmp.code==10 or icmp.code==13))

Null Scan

In this scan attacker sends a TCP packet without setting any flag on it and as a response if he is getting RST packet it means the port is closed.

udp.pcapng [Wireshark 1.10.2 (SVN Rev 51)	934 from /trunk-1.10)]					
Eile Edit View Go Capture Analyze S	tatistics Telephony Iools Inte	rnals <u>H</u> elp				
0 0 🛋 🔳 🔬 🗎 🗎 💥 😂	् 🗢 🗢 🖓 🛧 👱 🔳	III Q Q Q 🗹 I 🛛	K 🖾 🥵 % 🕻	1		
Filter: icmp.type==3 and icmp.code==3		Expression Clear Apply Sav	/e			
No. Time Source 279 27.5570940 192.168.145.12	Destination 9 192.168.145.131	Protocol Length Info ICMP 70 Destinat	ion unreachabl	e (Port unreachable)		
283 27.5580340 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	e (Port unreachable)		
287 27.5588950 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	(Port unreachable)		
292 27. 5630210 192. 168. 145. 12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	(Port unreachable)		
301 27 5683220 192 168 145 12	9 192.100.145.131	TCMP 70 Descript	ion unreachabl	(Port unreachable)		
306 27 6567730 192 168 145 12	9 192 168 145 131	TCMP 70 Destinat	ion unreachabl	(Port unreachable)		
308 27, 6597870 192, 168, 145, 12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	(Port unreachable)		
310 27,6601930 192,168,145,12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	(Port unreachable)		
312 27.6606560 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	(Port unreachable)		
314 27.6609830 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	(Port unreachable)		
316 27.6613320 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	e (Port unreachable)		
318 27.6617690 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	e (Port unreachable)		
320 27.6621020 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	<u>e (Port unreachable)</u>		
322 27.6624160 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	(Port unreachable)		
324 27.6659230 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	e (Port unreachable)		
326 27.6662/10 192.168.145.12	9 192.168.145.131	ICMP 70 Destinat	ion unreachabl	(Port unreachable)		
328 27.000/240 192.108.145.12	9 192,108,145,131	TCMP 77 Destinat	ion unreachabl	(Port unreachable)		
222 27 6674120 192 168 145 12	9 192.100.145.151	TCMP 70 Destinat	ion unreachabl	(Port unreachable)		
334 27 6677690 192 168 145 12	9 192.108.145.131	TCMP 70 Destinat	ion unreachabl	(Port unreachable)		
336 27 6681310 192 168 145 12	9 192 168 145 131	TCMP 70 Destinat	ion unreachabl	(Port unreachable)		
m Frame 279: 70 bytes on wire (5	60 bits), 70 bytes capt	ured (560 bits) on inte	erface 0			
	:3f (00:0c:29:40:e9:3f)	, Dst: Vmware_ca:3d:3e	(00:0c:29:ca:	3d:3e)		
Internet Protocol Version 4, S ■	rc: 192.168.145.129 (19	2.168.145.129), Dst: 1	92.168.145.131	(192.168.145.131)		
Finternet Control Message Proto	col					
Type: 3 (Destination unreach	able)					
Code: 3 (Port unreachable)						
Checksum: OnalGe [correct]						
Internet Protocol Version 4,	Src: 192.168.145.131 (192.168.145.131), Dst:	192.168.145.1	29 (192.168.145.129)		
i User Datagram Protocol, Src	Port: 60437 (60437), DS	t Port: 56141 (56141)				
0000 00 0c 29 ca 3d 3e 00 cc 2 0010 00 38 8e 50 00 00 80 01 0 0020 91 83 03 03 03 1 6c 00	9 40 e9 3f 08 00 45 00 8 1f c0 a8 91 81 c0 a8 0 00 45 00 00 1c fe 00 1 83 c0 a8 91 81 ec 15).=>)@.?E. .8.P 1E .9.z. .M5				
🛑 💅 Frame (frame), 70 bytes	Packets: 8173 · Displayed: 99	95 (12.2%) · Load time: 0:01.090			Profile: Default	
Figure 7. UDP Scan						

protocol_scan.pcapng [Wireshark 1.10.2 (SVN Rev	v 51934 from /trunk-1.10)]				
Eile Edit View Go Capture Analyze Statistic	s Telephony <u>I</u> ools Internals <u>H</u> elp				
• • 💉 🔳 🔬 🗁 🛗 💥 🔁 🔍 •	🗢 🔶 🛜 🛃 🔲 🗐 🖸	. २, २, 🖸 । 🔐 🗹 🥵 % ।	1		
Filter: icmp.type==3 and icmp.code==2	Expression	. Clear Apply Save			
No. Time Source	Destination Protocol	Length Info			
330 3.25607800 192.168.145.2	192.168.145.131 ICMP	62 Destination unreachab	le (Protocol unreachable)		
331 3.25608400 192.168.145.2	192.168.145.131 ICMP	62 Destination unreachab	ole (Protocol unreachable)		
332 3.25609100 192.168.145.2	192.168.145.131 ICMP	62 Destination unreachab	ole (Protocol unreachable)		
344 3.25617200 192.168.145.2	192.168.145.131 ICMP	62 Destination unreachat	ole (Protocol unreachable)		
345 3.25617800 192.168.145.2	192.168.145.131 ICMP	62 Destination unreachat	ole (protocol unreachable)		
346 3.25618500 192.168.145.2	192.168.145.131 ICMP	62 Destination unreachat	Ole (Protocol unreachable)		
347 3.25619200 192.168.145.2	192.168.145.131 ICMP	62 Destination unreachat	(Protocol unreachable)		
348 3.25019900 192.108.145.2	192.108.145.131 ICMP	62 Destination unreachat	(Protocol unreachable)		
349 3.23020000 192.108.145.2	102.168.145.131 ICMP	62 Description unreachat	(protocol unreachable)		
251 2 25621000 192 168 145 2	192.100.145.131 ICMP	62 Destination unreachad	(Protocol unreachable)		
352 3 25622600 192 168 145 2	102 168 145 131 TCMP	62 Destination unreachat	ale (Protocol unreachable)		
352 3 25622000 192 168 145 2	102 168 145 131 TCMP	62 Destination unreacha	ale (Protocol unreachable)		
354 3.25623900 192.168.145.2	192,168,145,131 TCMP	62 Destination unreachat	le (Protocol unreachable)		
367 3, 25634200 192, 168, 145, 2	192,168,145,131 TCMP	62 Destination unreachab	le (Protocol unreachable)		
368 3,25644400 192,168,145,2	192.168.145.131 ICMP	62 Destination unreachab	le (Protocol unreachable)		
260 2 25645100 102 168 145 2	103 168 145 121 TCMD	62 Portination unroachal	la (Brotocol uproschable)		
 Internet Control Message Protocol. Type: 3 (Destination unreachable) Code: 2 (Protocol unreachable) Charksum: 0xfcfd [correct] Internet Protocol version 4, Src Version: 4 Header length: 20 bytes Differentiated Services Field: Total Length: 20 Identification: 0xcb&e (52110) Hags: 0x00 Fragent offset: 0 	:: 192.168.145.131 (192.168. 0x00 (DSCP 0x00: Default;	145.131), Dst: 192.168.145. ECN: 0x00: Not-ECT (Not ECM	2 (192.168.145.2) H-Capable Transport))		
Time CO 11Ve: 58 Protocol: SATNET EXPAK (64) 0000 00 0C 29 Ca 3d 3e CO 50 56 f3 0010 00 30 ff 84 00 00 80 01 97 71 0020 91 83 03 02 fc fd 00 00 00 00 0030 00 00 3a 40 11 45 cO a8 91 83	ch 61 08 00 45 00).⇒> co a8 91 02 co a8 .0 45 00 00 14 cb 8e co a8 91 02:@.E	.P VaE. .q			
File: "I:\Wireshark Capture Files\protocol_sc	Packets: 2054 - Displayed: 254 (12.4%) -	Load time: 0:00.068		Profile: Default	

Figure 8. Protocol Scan



There will be no response to null scan if the port is open or filtered and if he is getting ICMP Type 3 Code 1,2,3,9,10 or 13 packet then *port seems to be firewalled.To detect Null Scan in Wireshark, we can use a simple filter* TCP.flags==0x000. It will filter all TCP packets moving without Flag (Figure 5).

Xmas Scan

Here the attacker sends packet with FIN, PSH & URG TCP flags and response is exactly the same like Null Scan. To detect this type of scan in Wireshark we can use filter "tcp.flags==0x029 (Figure 6).

ARP Poisoning-pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]	New York Concerning of Land	
File Edit View Go Capture Analyze Statistics Telephony Tools	Internals Help	
◎ ◎ 🖌 ■ 🔬 🗁 🕾 😂 🔍 ◆ ◆ 🐳 💈	■ ■ ♀ ♀ ◎, 四 ₩ ⊠ 🥵 ※ 第	
Filter: arp	Expression Clear Apply Save	
No. Time Source Destination 2 & 551247 Vmware_a0:32:5f Broadcast 3 & 551655 Vmware_a5:0d:e8 Vmware_a0:32:5f 12 13.56430 Vmware_a0:32:5f Vmware_a0:32:5f 2 13.564351 Vmware_a0:32:5f Vmware_a0:32:5f 2 13.6792108 Vmware_a0:32:5f Vmware_a0:32:5f 2 2 36.792108 Vmware_a0:32:5f Vmware_a0:32:5f 2 2 66.797105 Vmware_a0:32:5f Vmware_a0:32:5f 2 Coloring Rule String: arpl Ethernet II. Src: Vmware_a0:32:5f (00:0c:29:a0:32:5f) B costination: Vmware_a0:32:5f 00:0c:29:a0:32:5f 9 costination: Vmware_a0:32:5f (00:0c:29:a0:32:5f) 9 costination: Vmware_a0:32:5f (00:0c:29:a0:32:5f)	Protocol Length Info ARP 42 who has 192.168.10.37 Tell 192.168.10.2 ARP 60 192.168.10.3 is at 00:00:29:a5:0d:e8 ARP 60 who has 192.168.10.27 Tell 192.168.10.3 ARP 42 192.168.10.2 is at 00:00:29:a0:32:5f ARP 60 who has 192.168.10.27 Tell 192.405.10.3 (duplicate use ARP 42 192.168.10.2 is at 00:00:29:a0:5:5f (duplicate use of ARP 60 192.168.10.3 is at 00:00:29:e2:8e:55 ARP	of 192.168.10.3 detected!) 192.168.10.3 detected!)
■ Dop Texte address address of IP address: 131 ■ [Expert Info (warn/Sequence): Duplicate IP address: 131 [Expert Info (warn/Sequence): Duplicate IP address: 131 [Severity leve]: warn] [Group: sequence] [Seconds since earlier frame seen: 24] ■ Address Resolution Protocol (request) Hardware type: Ethernet (1) Protocol type: IP (0x0800) Hardware size: 6 Protocol size: 4 Opcode: request (1) sender Mac address: 192.168.10.3 (192.168.10.3) Target IP address: 192.168.10.3 (192.168.10.2) Target IP address: 25 fon 0c. 20 az 8e 4 80 56 00	<pre>ress configured (192.168.10.3)] 92.168.10.3)] :8e:84) 00:00:00) 01</pre>	E
0000 00 00 06 04 00 10 00 02 9 62 86 84 08 06 00 0010 08 00 06 04 00 10 00 02 99 62 86 84 08 06 00 0020 00 00 06 04 00 00 00 00 a8 0a 62 00 00 00 00 0030 00 00 00 00 00 00 00 00 00 00 00 00		
File: "C\Ulser\RhueRird\Downloads\Doan\Doan\Doan\Doan\Doan\Doan\Doan\Doan	8 (22 5%) - Load time 0:00 002	Profile: Default

Figure 9. ARP Poisioning

ARP Poisoning.pcap [Wireshar	k 1.10.2 (SVN Rev 51934 from /trunk-1.10)]		
Eile Edit View Go Capture	Analyze Statistics Telephony Tools	Internals Help	
0 0 🖌 🔳 🔬 🗎 🖴 🖿	a 🗶 🔁 🔍 🗢 🌳 🐢 🐺 👱	E E Q Q Q E B K 5 % B	
Filter: arp		Expression Clear Apply Save	
No. Time Source 2 8. 551247 Vimware 3 8. 551655 Vimware 12 13. 564303 Vimware 13 13. 564303 Vimware 20 36. 792108 Vimware 21 36. 792108 Vimware 22 36. 792655 Vimware 32 66. 797105 Vimware	Destination e_a0:32:5f Broadcast e_a5:0d:e8 Vmware_a0:32:5f e_a0:32:5f Vmware_a0:32:5f e_a2:8e:84 Vmware_b15	Protocol Length Info ARP 42 Who has 192.168.10.3 Tell 192.168.10.2 ARP 60 192.168.10.3 is at 00:00:29:a5:00:e8 ARP 60 Who has 192.168.10.2 Tell 192.168.10.3 ARP 60 hop has 192.168.10.2 Tell 192.168.10.3 ARP 60 192.168.10.2 is at 00:00:29:a0:32:5f ARP 60 192.168.10.2 is at 00:00:29:a0:32:5f ARP 60 192.168.10.3 is at 00:00:29:a0:32:5f ARP 60 192.168.10.3 is at 00:00:29:a0:32:5f	68.10.3 detected!) D.3 detected!)
Coloring Rule Stri Ethernet II, Src: Vmware, Bostination: vmware, Source: Vmware,e2:8 Type: ARP (0x0806) Padding: 000000000 Couplicate IP address: Efferame showing earl Expert Info (Warn [Expert Info (Warn [Seconds since earl [Group: Sequenc [Seconds since earl Address Resolution Pr Hardware type: Lthe Protocol size: 6 Protocol size: 6 Protocol size: 6 Protocol size: 5 Sender IP address: Sender Mac address: Sender Mac address: Sender IP address: Sander Size: Mac address: Sander Siz	ng: arp] are e2:8e =0:32:55 6ross 1(11) Warning:1 Group 4 Protocol 10 Sequence Cate IP a : warn] ie] ier frame totocol (r rrnet (1) Wmware_e 192.168.10.3 (192.168.10.3) 00:00:00.000 (00:00:00:	(4) Notes: 0 (0) Chats: 0 (0) Details: 15 Packet Comments: 0 • Sugmany • Count • Duplicate: IP address configured (192,166,10,5) • •	
0000 00 0c 29 a0 32 Sf 0010 08 00 66 04 00 01 0020 00 00 00 00 00 00 00 0030 00 00 00 00 00 00 00	00 0.c 29 e2 8e 84 08 06 00 0 0.0 0.c 29 e2 8e 84 c0 a8 0.a c c0 a8 0.a 02 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
File: "C:\Users\BlueBird\Dor	wnloads\Pcap\P Packets: 34 · Displayed:	3 (23.5%) + Load time: 0:00.002	Profile: Default

Figure 10. ExpertInfo window



UDP Scan

In UDP scan attacker sends a UDP packet (contains no meaningful data) on the target port and if that target responds with ICMP Type 3 Code 3 port is unavailable but if there is no response then it might be open or filtered. After capturing packets in Wireshark if you are getting high no. of packets with ICMP type 3 Code 3, it is a sign of UDP Scan. We can use filter icmp.type=3 and icmp.code=3 to detect UDP scan in Wireshark.

IP Protocol Scan

IP Protocol Scan is helpful in finding out protocols running over IP. To detect this attacker sends packet with different protocol nos., if he gets ICMP type 3 Code 2 Packet as a response then it means that this protocol is not running on the target system while no response means protocol is there or filtered. To detect this scan in Wireshark, we can apply icmp. type==3 and icmp.code==2 as a filter (Figure 8).

ARP Poisoning

ARP poisoning is a layer 2 redirection technique which can be easily identified by Wireshark. If more than one MAC addresses claim to have the same IP address it will highlight that packet as *Duplicate IP Address Detected* (Figure 9).

If we find challenge in finding such packets, by reading packet details we can go to top menu Analyze -> ExpertInfo and then Warnings tab as

shown in the picture (Figure 10). Here it will display all warning messages related to this capture which will help us to identify problems quickly.

Application Mapping

Wireshark can be used for application mapping as well, for example, if I am using HTTP communication then start looking for GET packet, within this packet if I will look for user-agent under Hypertext Transfer Protocol section it may reveal application OS and browser information (Figure 11). As we can see in the above picture that host is using Dropbox client tool version 2.0.22 on Windows 7 Operating System. I hope this article was useful and it will help you in understanding how Wireshark can be used to detect/analyze scanning traffic.

SANTOSH KUMAR



Santosh Kumar has more than 8 years of experience in IT Security. He is currently working as Technical Manager (IT Security) with Koenig Solutions Ltd. Santosh is proudly certified with Check Point Certified Managed Security Expert (CCMSE), Check Point Certified Security Expert (CCSE), CIS-CO ASA Specialist, Certified Ethical Hacker

(CEH) along with many others. He also has proven track record of streamlining security processes, design and implement efficient security solutions, lead and assist multi-disciplined, multinational teams in achieving security efficiency.

demo.pcapng (Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10))	
Eile Edit Yiew Go Capture Analyze Statistics Telephony Iools Internals Help	
● ● 🖌 ■ 🔬 🗁 🔍 ♦ ♦ 🗢 7 🖢 🔲 🖃 이 이 이 이 🖓 🖄 🐯 🕺 🔛	
Fitter: http://www.clear.Apply_Save	
No. Time Source Destination Protocol Length Info 172 17.5482040108.160.163.46 192.168.1.2 ΗΤΤΡ 233 ΗΤΤΡ 170 172 17.5523170192.168.1.2 108.160.163.46 HTTP 382 GFT /subscribe?host_int=758881595& map=243450228_111912599924,257433	8298_26027242074,243453548_17204
<pre>Livream rnuex. oj Sequence number: 1 (relative sequence number) [Next sequence number: 329 (relative sequence number)] Acknowledgment number: 180 (relative sequence number)] Acknowledgment number: 180 (relative ack number) Header length: 20 bytes B Flags: 0x018 (PSH, AcK) Window size value: 4191 [calculated window size: 4191] [calculated window size: 4191] [chalculated window size: 4191] [sepert Info (chal/Sequence): GET /subscribe?host_int=75881595&ns_map=243450228_111912599924,257438298_26027242074,243453548_172042145388&user_id=1517 [severity level: chal] [chalculated window comp: 5 statistic for 7 statisti</pre>	s=1381657740 HTTP/l.1\r\n 145388&user_id=151764527&nid=2 764527&nid=2&ts=1381657740 HTT 527&nid=2&ts=1381657740 ■ 48_172042145388&user_id=151764
[HTTP request 2/2]	
0000 0c d2 b5 01 8b 3b 84 a6 c8 7e 11 ca 08 00 45 00	-
Image: Straine (frame), 382 bytes Packets: 210 · Displayed: 2 (1.0%) · Load time: 0:00.015 Prof	file: Default

Figure 11. Application Mapping



Be Certified **Expert in**





www.koenig-solutions.com

+91 88009 40444

inux

Web Apps