Abril 2018 Volumen 1, nº 1

Departamento de Seguridad Informática MINSAP



Contenido:

ANÁLISIS DE TRÁFICO CON WIRESHARK Parte II

1

ANÁLISIS DE TRÁFICO CON WIRESHARK Parte II

3. DÓNDE REALI-ZAR LA CAPTURA DE DATOS

El primer paso para poder auditar la red será definir dónde analizar el tráfico. Imaginemos un escenario común. Nos encontramos en un entorno conmutado formado por varios switches, unos cuantos equipos y un servidor de ficheros. El rendimiento de la red ha disminuido en los últimos días y desconocemos la causa.

Carecemos de un IDS que pueda dar la voz de alarma sobre algún ataque o anomalía en la red y sabemos que el servidor de ficheros abastece, en cuanto a tasa de transferencia se refiere. a los equipos de nuestra LAN (Local Area Network) sin problema alguno. Además, nuestros equipos de red no cuentan con protocolos como Netflow para poder analizar tráfico remotamente por lo que decidimos utilizar Wireshark. La primera duda que surge es dónde instalarlo.

A pesar de parecer lógico instalar Wireshark en el propio servidor de ficheros para analizar el tráfico que transita por ese segmento de red, nos encontraremos con situaciones en las cuales no podamos tener acceso físico al servidor o simplemente, por motivos de seguridad, por ejemplo entornos SCA-DA, no podamos instalar software en el mismo.

En este caso se mostrarán algunas alternativas en el uso de técnicas que permitan llevar a cabo una captura de tráfico sin necesidad de portar Wireshark al propio servidor. La excepción a esta regla la veremos en el último caso, donde se proponen varios métodos de captura remota en los que sí es necesario ejecutar o al menosinstalar aplicaciones en el equipo que se quiere monitorizar.

3.1. UTILIZANDO UN HUB

Si conectásemos un equipo con Wireshark a uno de los puertos del switch, solo veríamos las tramas que transcurren entre el switch y nuestra máquina, y eso no es lo que pretendemos. El switch divide la red en segmentos, creando dominios de colisión separados y eliminando, de esta forma, la necesidad de que cada estación compita por el medio. Únicamente envía las tramas a todos los puertos (pertenecientes a la misma VLAN) cuando se trata de difusiones broadcast (por ejemplo, para saber la dirección física de alguna máquina).

Una de las alternativas que tenemos para alcanzar nuestro propósito es hacer uso de un hub, como se aprecia en la Figura 1-Modos de captura y conectarlo en el mismo segmento de red donde se encuentra nuestro servidor. Al tratarse ahora de un medio compartido, todo el tráfico entre el switch y el servidor podrá analizarse en nuestro equipo.

3.2. PORT MIRRO-RING O VACL (VLAN -BASED ACLS)

Siempre que tengamos acceso al switch, y soporte esta funcionalidad, será la manera más cómoda para capturar el tráfico de red. Dicho modo de trabajo, denominado modo SPAN en entornos Cisco, permite duplicar el tráfico que transcurre por uno o varios puertos del switch y replicarlo al puerto que queramos. Hay que tener en cuenta que el puerto configurado como mirroring tiene que ser tan rápido como el puerto/puertos a monitorizar para evitar pérdida de tramas. Este método es empleado por muchos administradores para instalar IDS u otras herramientas de monitorización.

Una ventaja que presentan las VACL frente al Port Mirroring es que permiten una mayor granularidad a la hora de especificar el tráfico que se quiere analizar. Mientras que configurando Port Mirroring es posible redirigir el tráfico de un puerto o VLAN a otro, con VACL es posible especificar ACLs para seleccionar el tipo de tráfico en el que estamos interesados

En el siguiente ejemplo, se define una VLAN Access Map para reenviar y capturar paquetes que coincidan con el tráfico definido en lab_10 y que posteriormente será aplicado a las VLANS 14,15 y 16:

Router(config)# vlan access-map bmf 10

Router(config-access-map)# match ip address lab_10

Router(config-access-map)# action forward capture

Router(config-access-map)# exit

Router(config)# vlan filter bmf vlan-list 14-16

Router# show ip access-lists lab_10

Extended IP access list lab_10

permit ip 10.0.0.0 0.255.255.255 any

Algunos dispositivos Cisco también disponen de una funcionalidad denominada MiniProtocol Analyzer gracias a la cual se puede capturar tráfico desde una sesión SPAN y almacenar los paquetes en un buffer local, pudiendo ser posteriormente exportados en un fichero .cap. Esta funcionalidad también permite especificar opciones de filtrado para limitar la captura de paquetes, por ejemplo, podrían especificarse aquellos paquetes que tengan un EtherType determinado o aquellos declarados en una ACL previamente configurada. Además, utiliza libpcap como formato de captura por lo que puede emplearse Wireshark o cualquier otro analizador de protocolos para un análisis posterior.

3.3. MODO BRIDGE

En caso de no tener acceso al switch, podremos utilizar un equipo con dos tarjetas de red para situarnos entre el switch y el servidor, como se observa en la Figura 1.

Consiste en un MitM (Man in the Middle), a nivel físico, donde tendremos un acceso pasivo a todo el caudal de tráfico.

Tenemos varias alternativas para poner nuestro PC en este modo de funcionamiento, pero destacamos las bridge-utils (paquete de utilidades bridge para Linux) por su facilidad de instalación y configuración. Únicamente tendremos que crear una interfaz de tipo bridge y posteriormente añadir las interfaces físicas que forman parte de dicho puente. Por último, levantaremos la interfaz y ejecutaremos Wireshark. El inconveniente de éste método de captura es la pérdida de tramas durante su instalación, situación que en ciertos escenarios no es asumible. A continuación, se muestra un ejemplo de su configuración:

root@bmerino:~# brctl addbr mybridge

root@bmerino:~# brctl addif mybridge
eth1

root@bmerino:~# brctl addif mybridge eth0

root@bmerino:~# ifconfig mybridge up

3.4. ARP SPOOF

En contadas ocasiones, y en los casos en los que no podamos utilizar los métodos anteriores, podemos hacer uso de herramientas como Ettercap o similares para llevar a cabo un MitM (Man in the Middle). Es importante entender que se trata de un método bastante ofensivo y que únicamente será útil en entornos no críticos, donde prima cierta necesidad en interceptar tráfico entre varias máquinas.Lo que conseguiremos será que el equipo que se desea monitorizar envíe todas las tramas a través de nuestro PC donde tendremos Wireshark ejecutándose. El proceso se lleva a cabo contaminando la cache de los equipos involucrados con una asociación IP/MAC falsa. Algunos switches disponen de funcionalidades que les permiten detectar este proceso Dynamic Arp Inspection y (véase DHCP Snooping), por lo que es importante deshabilitar dicha funcionalidad en los dispositivos de red si no queremos que nuestro puerto entre en modo shutdown. Para interponernos entre el servidor (10.0.0.100) y el gateway de nuestra LAN (10.0.0.1) bastará con ejecutar Ettercap de la siguiente forma:

root@bmerino:~# ettercap -T -M arp:remote /10.0.0.1/ /10.0.0.100/ &



3.5. REMOTE PACKET CAP-TURE

Además de los métodos citados anteriormente, existen varias posibilidades para capturar datos de forma remota. Una de ella es mediante RPCAP (Remote Packet Capture System), aunque en este caso sería necesario ejecutar un programa servidor (rpcapd) junto con las librerías necesarias en el equipo a monitorizar y un programa cliente desde el cual se recuperarán y visualizarán los mismos; en nuestro caso, Wireshark.

Como hemos dicho anteriormente, este método es apropiado para entornos no críticos donde tenemos posibilidad de instalar software en el equipo cuyo tráfico queremos analizar, con el riesgo que ello conlleva para la estabilidad y rendimiento del mismo.

Para la configuración del servidor, únicamente hay que ejecutar rpcapd.exe, incluido en la instalación de WinPcap 4.0 (librerías libpcap en equipos Windows) o superior.

Se puede especificar el puerto de

escucha y otras opciones como autenticación, lista de clientes autorizados a conectar al servidor, etc. El modo de funcionamiento puede ser activo o pasivo. En el primer caso el demonio tratará de establecer una conexión hacia el cliente para que éste envíe los comandos adecuados al servidor. Este modo de funcionamiento será útil cuando el demonio esté detrás de un Firewall que no tenga NAT configurado para su conexión desde el exterior. En el segundo caso, será el cliente el que inicie la conexión con el servidor para comenzar a monitorizar datos. (Imagen 1)

El cliente tendrá que especificar dirección, puerto, credenciales (en el caso de que así fuera requerido por el servidor) y la interface desde la cual se desean capturarpaquetes. En Wireshark, esto se realiza desde Capture >> Options y especificando en Interface el tipo Remote: (Imagen 2)

Es importante destacar que, si la captura se realiza en la misma interfaz en la que se está utilizando el propio protocolo RPCAP para transferir los datos entre el demonio y el cliente, dichos paquetes también serán visualizados en Wireshark pudiendocomplicar la interpretación de los mismos. Se puede impedir que estos paquetes interfieran con el resto. Para ello, tendremos que seleccionar la opción "Do not capture own RPCAP traffic" dentro de "Remote Settings".

Otra alternativa aparte de RPCAP para la captura remota de datos es redirigir la salida de tcpdump desde una conexión ssh. Lógicamente, en este caso el equipo a monitorizar necesita disponer de acceso ssh y tener tcpdump instalado (Imagen 3)

Una vez configurada nuestra máquina, haciendo uso de cualquiera de los métodosanteriores, podemos lanzar Wireshark como root/administrador. Para iniciar la capturaseleccionamos la interfaz en el menú Capture >> Interfaces (en el caso de optar por el uso del modo bridge, podemos utilizar cualquiera de las dos). (Imagen 4)

A continuación, describimos brevemente las áreas más interesantes que nos muestra Wireshark según comienza la toma de datos (Figura 5- Áreas de Wireshark): La zona 1 es el área de definición de filtros y, como veremos más adelante,

permite definir patrones de búsqueda para visualizar aquellos paquetes o

protocolos que nos interesen.

La zona 2 se corresponde con la lista de visualización de todos los paquetes

que se están capturando en tiempo real. Saber interpretar correctamente los datos proporcionados en esta zona (tipo de protocolo, números de secuencia, flags, marcas de tiempo, puertos, etc.) nos va a permitir, en ciertas ocasiones,deducir el problema sin tener que realizar una auditoría minuciosa.

La zona 3 permite desglosar por capas cada una de las cabeceras de los paquetes seleccionados en la zona 2 y nos facilitará movernos por cada uno de los campos de las mismas. Por último, la zona 4 representa, en formato hexadecimal, el paquete en bruto, es decir, tal y como fue capturado por nuestra tarjeta de red. .

Imagen 1: Captura de datos con rpcapd

C:\Program	Files\WinPcap>rpcapd.exe -n	-p 8086
Press CTRL	+ C to stop the server	

Imagen 2: Conexión a servidor rpcapd

Г	Capture					
	Interface: Re	emote	✓ Conexi⊠n de	red Intol/P) DE	O/1000 MTL Device) NI	
	IP address: fe	80::3dc9:6e	e5:6b87:18e2, 192.1	168 🗖 Wires	hark: Remote Int 👝	
	Link-laver he	ader type	Ethernet 💌	Host:	192,168,1,200	
		ader type.	culence ·	11050	152:100:1:200	
	Capture p	ackets in p	romiscuous mode	Port:	6666	
	Capture packets in pcap-ng f		cap-ng format (ex	per Authent	Authentication	
	Imagon 2: tor	dumn				
	iniagen 5. tcp	Juunip				
	root@boriaB]	[:∼# ssh r	oot@192.168.254	4.211 tendum	n -w - 'nort 122'	wireshark -k -i
	root@192.168	3.254.211'	s password:	+.211 cepuun	ip -w - port :22	WITESHARK -K -I
	tcpdump: lis	stening on	eth0. link-tv	pe EN10MB (E	thernet), capture s	ize 96 bytes
	cepuump: cri	scening on	ceno, erink ey		cherner, capture s	120 50 bytes
			- uli			
	Imagen 4: Are	eas de wiresh	ark			
	File Edit View G	o <u>Capture</u> Anal	yze <u>S</u> tatistics <u>H</u> elp			
				4 - 3 Z		
1						
1	Eilter:			Expression.	🛃 Limpiar 🖌 Aplicar	
1	Filter:	Source	Destination	Expression. Protocol	Jimpiar Aplicar	
1	Eiter:	Source	Destination 239,255,255	Protocol	Info	
1	Filter: No Time 4 9.028195 5 9.678865 6 9.681088	Source	Destination 239.255.255 e:a2:69 Broadcast p:72:04 IntelCor_6e	Protocol .250 SSDP .4PP :a2:69 APP	Limpiar Aplicar Into M.SEARCH + HTTP/1.1 Who has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7	0.0.101
1	Filter: No Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034	Source 10.0.0.109 IntelCor_6 Cisco-Li_2 IntelCor_6	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast	Protocol .250 SSDP .a2:69 APP APP	Limpiar Aplicar Aplicar M·SEARCH + HTTP/1.1 Who has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10	0.0.101 /2:04 0.0.0.101
1	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10 76817	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_42 2 10 0 100	Destination 239.255.255 e:a2:69 Broadcast p:72:04 IntelCor_6e e:a2:69 Broadcast 9:bd:93 IntelCor_6e	Protocol 250 SSOP ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP	Limpiar Aplicar Info M·SEARCH + HTTP/1.1 Who has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo. (inc) request	0.0.101 /2:04 0.0.0.101 9:bd:93
1	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 10 10.80007	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_48 2 10.0.0.100 2 10.0.0.1	Destination 239.255.255 e:a2:69 Broadcast o:72:04 IntelCor_6e e:a2:69 Broadcast 9:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100	Protocol .250 SSDP ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP ICMP ICMP	↓ impiar ▲ Aplicar Info M-SEARCH + HTTP/1.1 Who has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:49 Echo (ping) request Echo (ping) request	0.0.101 /2:04 0.0.0.101 9:bd:93
1	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_48 2 10.0.0.100 2 10.0.0.1 6 IntelCor_66	Destination 239.255.255 2:32:69 Broadcast 2:72:04 IntelCor_6e 2:32:69 Broadcast 2:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 2:32:69 Cisco-Li_2b	Protocol 250 SSDP ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP	Limpiar Aplicar Aplicar M.SEARCH + HTTP/1.1 Who has 10.0.0.1? Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:66	0.0.101 2:04 0.0.0.101 9:bd:93
1	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 12 10.80024	Source 10.0.0.109 IntelCor_66 Cisco-Li_22 IntelCor_66 IntelCor_48 2 10.0.0.100 2 10.0.0.1 6 IntelCor_66 5 IntelCor_66	Destination 239.255.255 2:2:69 Broadcast 2:72:04 IntelCor_6e 2:2:69 Broadcast 2:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 2:a2:69 Cisco-Li_2b 2:a2:69 IntelCor_49 2:a2:69 IntelCor_49	Protocol Protocol Protocol Protocol APP APP A22:69 APP ICMP ICMP ICMP ICMP id2:04 APP	Limpiar Aplicar Into M.SEARCH + HITP/1.1 Who has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:49 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e 10.0.0.1 is at 00:13:ce:6e	0.0.101 /2:04 0.0.0.101 9:bd:93 9:a2:69 9:2:69
1 2	No Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045	Source 10.0.0.109 IntelCor_66 Cisco-Li_22 IntelCor_66 IntelCor_42 2 10.0.0.100 2 10.0.0.1 6 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 0.0.0.100 2 10.0.0.100 3 0.0.00 3 0.0.00 3 0.0.00 3 0.0.00 3 0.0.00 3 0.00 3 0	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b b:a2:69 IntelCor_49 a:a2:69 Cisco-Li_2b		Limpiar Aplicar Aplicar M.SEARCH + HITP/1.1 Who has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:49 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:a2:69 0:a2:69
r [No Time 4 9.028195 5 9.678965 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80027 13 11.81045 14 11.83372 15 11.485725	Source 10.0.0.109 IntelCor_66 Cisco-Li_22 IntelCor_64 2 10.0.0.100 2 10.0.0.1 6 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_67 1 IntelCor_	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 b:a2:69 IntelCor_40 b:a2:69 IntelCor_40 b:a2:69 IntelCor_40 b:a2:60		Limpiar Aplicar Aplicar M.SEARCH + HTTP/1.1 Who has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request I0.0.0.100 is at 00:13:ce:6e 1300 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e 1300 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e 1300 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e 1300 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e 1300 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e 1300 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e 10.0.0.1 is at 0	0.0.101 72:04 0.0.0.101 9:bd:93 9:a2:69 9::a2:69 0 MSS=1460 72:69
2	No Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.85924 16 11.85924	Source 10.0.0.169 IntelCor_66 Cisco-Li_22 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 Broadcast		Limpiar Aplicar Aplicar Mr. SEARCH + HTTP/1.1 Who has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:49 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e: 10.0.0.10 is at 00:13:ce:6e: 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a who has 10.0.0.17 Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:a2:69 0:mSS=1460 12:69 0.0.101
2	No. Time 4 9.028195 5 9.678865 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.85924 1	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66	Destination 239.255.255 2:32:69 Broadcast 2:72:04 IntelCor_6e 2:32:69 Broadcast 3:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 2:32:69 Cisco-Li_2b 2:32:69 IntelCor_49 2:32:69 IntelCor_49 2:32:69 Broadcast	Protocol Protocol ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :CMP ICMP ICMP ICMP :72:04 ARP :72:04 ARP :0d:93 ARP :Dd:93 ARP :bd:93 ARP :bd:93 ARP	Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:49 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a who has 10.0.0.17 Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:a2:69 0:mss=1460 02:69 0.0.101
2	No Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.85924 14	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_66 10.0.0.100 2 10.0.0.100 2 10.0.0.100 3 IntelCor_66 1 IntelCor_66 4 10.0.0.100 7 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 7 IntelCor_66 6 IntelCor_66 7 IntelCor_66 7 IntelCor_66 8 IntelCor_66 8 IntelCor_66 8 IntelCor_66 9 IntelCor_66	Destination 239.255.255 e:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 e:a2:69 IntelCor_49 e:a2:69 IntelCor_49 e:a2:69 IntelCor_49 e:a2:69 Broadcast	Protocol Protocol ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :a2:69 ARP :CMP ICMP ICMP ICMP :72:04 ARP :72:04 ARP :0d:93 ARP :bd:93 ARP :bd:93 ARP	Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.101 is at 00:13:ce:6e:a 10.0.0.1 is at 00:13:ce:6e:a 10.0.0.1 is at 00:13:ce:6e:a 10.0.0.1 is at 00:13:ce:6e:a 10.0.0.1 is at 00:13:ce:6e:a Mo has 10.0.0.17 Tell 10.0	0.0.101 /2:04 0.0.0.101):bd:93 1:a2:69 1:a2:69 1:a2:69 0.055=1460 02:69 0.0.101
2	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.859244 14 Protocol size 2	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_48 2 10.0.0.100 2 10.0.0.1 6 IntelCor_66 5 IntelCor_66 4 10.0.0.100 7 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 7 IntelCor_66 6 IntelCor_66 7 IntelCor_66	Destination 239.255.255 2:32:69 Broadcast 2:72:04 IntelCor_6e 2:32:69 Broadcast 2:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 2:32:69 IntelCor_49 2:32:69 IntelCor_49 2:32:69 IntelCor_49 2:32:69 Broadcast	Protocol Protocol Protocol APP APP a2:69 APP ICMP ICMP	Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.100 is at 00:13:ce:6e:a 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a Who has 10.0.0.17 Tell 10.0	0.0.101 /2:04 0.0.0.101 2:69 2:69 2:69 0 MSS=1460 02:69 0.0.101
2	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.90007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.85924 14 11.83924 Hardware \$12 Protocol \$12 Opcode: repl	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_42 10.0.0.100 2 10.0.0.1 6 IntelCor_66 1 IntelCor_66 1 IntelCor_66 4 10.0.0.100 7 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 7 IntelCor_66 6 IntelCor_66 7 IntelCor_66 8 IntelCor_66 9	Destination 239.255.255 2:32:69 Broadcast 2:72:04 IntelCor_6e 2:32:69 Broadcast 2:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 2:32:69 IntelCor_49 2:32:69 IntelCor_49 2:32:69 IntelCor_49 2:32:69 Broadcast	Protocol Protocol Protocol APP a2:69 APP ia2:69 APP id2:69 APP id2:69 APP id2:69 APP id2:69 APP id2:93 id2:9 id2:93 id2:9 id2:93 id2:9 id2:93 id2:9 id2:93 id2:9 id2:	Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.100 is at 00:13:ce:6e:a 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a Who has 10.0.0.1? Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:a2:69 0:mss=1460 0:2:69 0.0.101
L	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.85924 4 Hardware \$12 Protocol \$12 Opcode: repl Sender MAC \$2	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_42 10.0.0.100 2 10.0.0.1 6 IntelCor_66 1 IntelCor_66 1 IntelCor_66 4 10.0.0.100 7 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 6 IntelCor_66 7 IntelCor_66 6 IntelCor_66 7 IntelCor_66 8 IntelCor_66 9	Destination 239.255.255 2:32:69 Broadcast 2:72:04 IntelCor_6e 2:32:69 Broadcast 2:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 2:32:69 Cisco-Li_2b 2:32:69 IntelCor_49 2:32:69 IntelCor_49 2:32:69 Broadcast 2:cor_49:bd:93 (00:12:f0:	Protocol Protocol Protocol APP APP APP APP COP IOP IOP TOP COP COP COP COP COP COP APP COP CO	Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.100 is at 00:13:ce:6e:a 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a Who has 10.0.0.1? Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:a2:69 0:mss=1460 0:2:69 0.0.101
1	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.859244 ************************************	Source 10.0.0.109 IntelCor_66 Cisco-Li_22 IntelCor_66 IntelCor_66 2 10.0.0.100 2 10.0.0.100 2 10.0.0.100 3 IntelCor_66 1 IntelCor_66 1 IntelCor_66 4 10.0.0.100 7 IntelCor_66 6 IntelCor_66 1 IntelCor_66 6 IntelCor_66 20: 4 10,00002) address: IntelCo 20: 5 20: 5	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 Broadcast b:a2:69 Broadcast cor_49:bd:93 (00:12:f0:40) cor_49:bd:93 (00:12:f0:40) b:a2:69 Broadcast	Protocol Protocol Protocol Protocol APP APP APP APP COP IOMP IOMP IOMP IOMP TOP COP COP COP COP APP APP COP COP APP APP COP C	Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.10 is at 00:13:ce:6e:a Mho has 10.0.0.1? Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:mss=1460 0:2:69 0.0.101
L	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.85924 1 Hardware s12 Protocol siz Opcode: repl Sender IP ac Sender IP ac Target MAC ac Target MAC ac	Source 10.0.0.109 IntelCor_66 Cisco-Li_22 IntelCor_66 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 2 I0.0.0.100 7 IntelCor_66 1 IntelCor_66 2 IntelCor_66 2 IntelCor_66 3 IntelCor_66 4 I0.0.0.100 7 IntelCor_66 5 IntelCor_	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 Broadcast b:a2:69 Broadcast cor_49:bd:93 (00:12:f0: 100 (10.0.0.100) cor_6e:a2:69 (00:13:ce:		Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:49 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e: 10.0.0.10 is at 00:13:ce:6e: 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a Who has 10.0.0.17 Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:mss=1460 12:69 0.0.101
1	No. Time 4 9.028195 5 9.678865 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.833722 15 11.85725 16 11.859244 1 Hardware sız Protocol siz Opcode: repl Sender IP ac Sender IP ac Target MAC ac Target IP ac	Source 10.0.0.109 IntelCor_66 Cisco-Li_24 IntelCor_66 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 2 I0.0.0.100 7 IntelCor_66 2 IntelCor	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 Broadcast b:a2:69 Broadcast cor_49:bd:93 (00:12:f0:4) cor_6e:a2:69 (00:13:ce:4) 101 (10.0.0.101)		Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:49 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.10 is at 00:13:ce:6e:a 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a Who has 10.0.0.17 Tell 10.0	0.0.101 /2:04 0.0.0.101 9:bd:93 9:a2:69 9::a2:69 0:MSS=1460 0:2:69 0.0.101
1	No Time 4 9.028195 5 9.678965 5 9.678965 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.859244 Protocol siz 0pcode: repl Sender MAC a Sender IP ac Target MAC a Target IP ac 00000 00 13 ce 66	Source 10.0.0.109 IntelCor_66 Cisco-Li_22 IntelCor_66 IntelCor_66 2 10.0.0.100 2 10.0.0.100 2 10.0.0.100 3 IntelCor_66 1 IntelCor_66 1 IntelCor_66 1 IntelCor_66 4 10.0.0.100 7 IntelCor_66 1 IntelCor_66 2 E: 6 2 E: 6 2 E: 4 Ly (0x0002) address: IntelCor_66 2 E: 4 Ly (0x0002) address: IntelCor_66 2 E: 6 2 E: 7 2 E: 7	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 Broadcast bor_49:bd:93 (00:12:f0: 100 (10.0.0.100) bor_6e:a2:69 (00:13:ce: 101 (10.0.0.101) 10 49 bd 93 08 06 00 0		Limpiar Aplicar Aplicar Moo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 Who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.10 is at 00:13:ce:6e:a 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a Who has 10.0.0.1? Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:mSS=1460 12:69 0.0.101
L	No Time 4 9.028195 5 9.678965 5 9.678965 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.85924 Hardware s12 Protocol si2 Opcode: repl Sender MAC a Sender IP ac Target MAC a Target IP ac 0000 00 13 ce 66 0010 08 00 00 00 13 ce 66 0010 08 00 00 00	Source 10.0.0.169 IntelCor_66 Cisco-Li_22 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 IntelCor_66 Ce: 6 Ce: 6 Ce: 6 Ce: 6 Ce: 7 Cores: IntelCor_66 Cores: IntelCor_66 Core 6 Ce: 7 Core 6 Ce: 7 Core 6 Core 7 Core 7 Co	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 Broadcast b:cor_49:bd:93 (00:12:f0: 100 (10.0.0.100) b:or_6e:a2:69 (00:13:ce: 101 (10.0.0.101) 10 49 bd 93 08 06 00 0 f0 49 bd 93 0a 00 06 6	▼ Expression. Protocol .250 ARP .32:69 ARP .22:69 ARP .22:69 ARP .22:69 ICMP .1CMP ICMP .22:04 ARP .4PP .22:04 .22:04 .22:04 .22:04 .22:04 .22:04 .22:04 .22:04 .22:04 <th>Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.1 is at 00:13:ce:6e:a 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a who has 10.0.0.1? Tell 10.0</th> <th>0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:a2:69 0:mSS=1460 0:2:69 0.0.101</th>	Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.1 is at 00:13:ce:6e:a 1390 > www [SYN] Seq=0 Len=0 10.0.0.1 is at 00:13:ce:6e:a who has 10.0.0.1? Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:a2:69 0:mSS=1460 0:2:69 0.0.101
3	No Time 4 9.028195 5 9.678965 5 9.678965 6 9.681088 7 9.692034 8 9.696736 9 10.76817 10 10.80007 11 10.80017 12 10.80024 13 11.81045 14 11.83372 15 11.85725 16 11.85924 Hardware s12 Protocol si2 Opcode: repl Sender IP ac Target IP ac 7 00000 00 13 ce 66 0010 08 00 00 04	Source 10.0.0.169 IntelCor_66 Cisco-Li_22 IntelCor_62 2 10.0.0.100 2 10.0.0.100 2 10.0.0.100 2 10.0.0.100 3 IntelCor_66 1 IntelCor_66 1 IntelCor_66 4 10.0.0.100 7 IntelCor_66 1 IntelCor_66 2 c: 6 2 c: 6 2 c: 4 Ly (0x0002) address: IntelCor 2 ddress: 10.0.0. 2 a2 69 00 12 4 00 02 00 12 4 a2 69 00 02 2 a2 69 00 00	Destination 239.255.255 a:a2:69 Broadcast b:72:04 IntelCor_6e a:a2:69 Broadcast b:bd:93 IntelCor_6e 10.0.0.1 10.0.0.100 a:a2:69 Cisco-Li_2b a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 IntelCor_49 a:a2:69 Broadcast Cor_49:bd:93 (00:12:f0:. 100 (10.0.0.100) cor_6e:a2:69 (00:13:ce: 101 (10.0.0.101) 10 49 bd 93 08 06 00 0 f0 49 bd 93 0a 00 06 00 65	▼ Expression. Protocol .250 ARP .32:69 ARP .22:69 ARP .22:69 ICMP ICMP ICMP ICMP :72:04 ARP :72:04 ARP :72:04 ARP :04:93 ARP :72:04 ARP :04:93 ARP .04:93 ARP .04:93 ARP .05:00 .00 .00 .00 .00 .00 .00 .00 .00 .00	Limpiar Aplicar Aplicar Mo has 10.0.0.17 Tell 10.0 10.0.0.1 is at 00:18:39:2b:7 who has 10.0.0.100? Tell 10 10.0.0.100 is at 00:12:f0:45 Echo (ping) request Echo (ping) request 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.100 is at 00:13:ce:6e:a 10.0.0.1 is at 00:13:ce:6e:a Who has 10.0.0.1? Tell 10.0	0.0.101 /2:04 0.0.0.101 0:bd:93 0:a2:69 0:a2:69 0:mSS=1460 12:69 0.0.101