

بسمه تعالی



گزارش کار سوم آزمایشگاه شبکه

## آشنایی پیشرفته با نرم افزار Wireshark، نحوه ی تنظیم DNS Server

استاد:

دکتر بردیا صفایی

نویسندگان:

امیررضا آذری 99101087

امیرمحمد صالح 99101824

بزرگمهر ضیا 99100422

دانشگاه صنعتی شریف

تابستان 1403

## فهرست

هدف .....	3
بخش اول _ Wireshark .....	3
1.1. به دست آوردن captcha .....	3
1.2. سوال ها .....	9
بخش دوم _ راه اندازی DNS .....	16
2.1. سناریو آزمایش .....	16
2.2. سوال ها .....	24
منابع و مراجع: .....	25

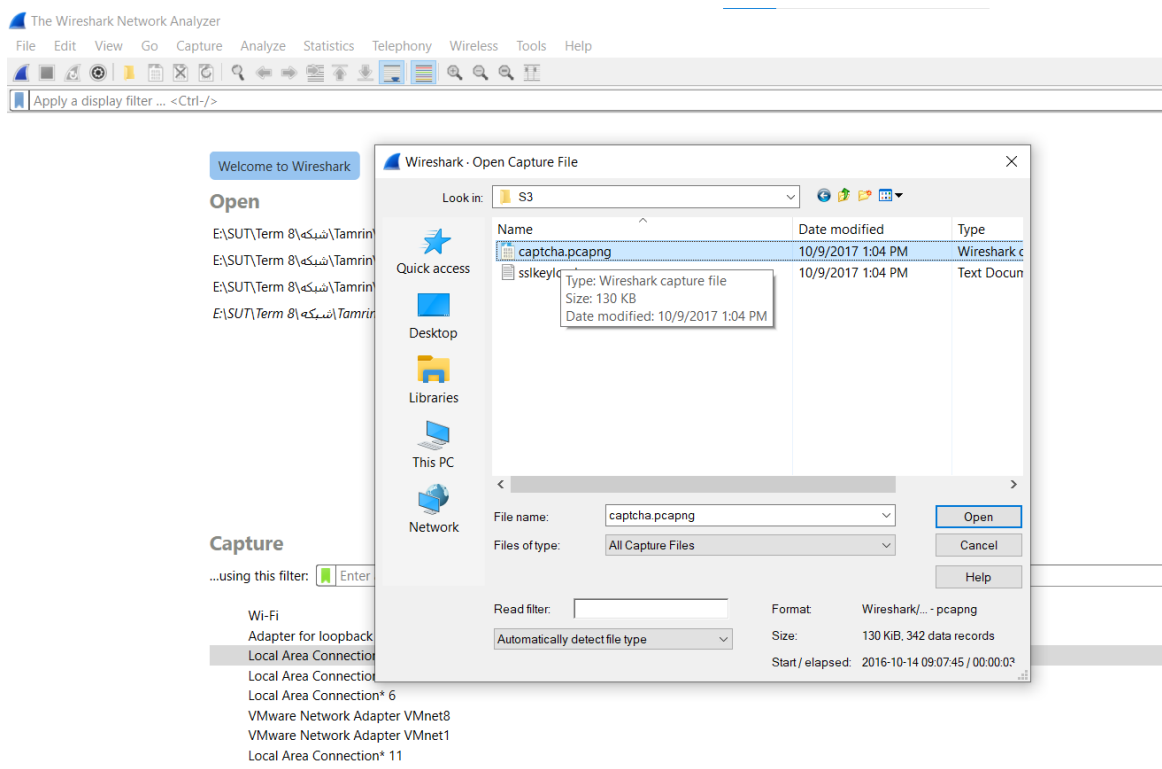
## هدف

در آزمایش قبلی، با نرم‌افزار Wireshark و همچنین DNS آشنا شدیم. با استفاده از این نرم‌افزار می‌توان ترافیک شبکه را تحلیل کرد. حال در بخش اول این آزمایش می‌خواهیم برخی از قابلیت‌های کاربردی این نرم‌افزار را بررسی نماییم. در بخش دوم نیز می‌خواهیم نحوه‌ی راه‌اندازی یک کارگزار DNS را نشان بدهیم.

## بخش اول \_ Wireshark

### 1.1. به‌دست آوردن captcha

ابتدا نرم‌افزار وایرشارک را باز می‌نماییم. سپس مطابق آنچه در دستور کار عنوان شده است، فایل captcha.pcapng را از طریق بخش File و زیربخش Open باز می‌کنیم.



تصویر 1. باز کردن فایل captcha.pcapng

بعد از باز کردن فایل مورد نظر در نرم‌افزار، با صفحه‌ی زیر مواجه می‌شویم:

Apply a display filter... <Ctrl> /

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.2	239.255.255.250	SSDP	306	NOTIFY * HTTP/1.1
2	0.102277	192.168.1.2	239.255.255.250	SSDP	315	NOTIFY * HTTP/1.1
3	0.205056	192.168.1.2	239.255.255.250	SSDP	378	NOTIFY * HTTP/1.1
4	0.307107	192.168.1.2	239.255.255.250	SSDP	370	NOTIFY * HTTP/1.1
5	0.409506	192.168.1.2	239.255.255.250	SSDP	315	NOTIFY * HTTP/1.1
6	0.511924	192.168.1.2	239.255.255.250	SSDP	354	NOTIFY * HTTP/1.1
7	0.614340	192.168.1.2	239.255.255.250	SSDP	386	NOTIFY * HTTP/1.1
8	0.716687	192.168.1.2	239.255.255.250	SSDP	315	NOTIFY * HTTP/1.1
9	0.819144	192.168.1.2	239.255.255.250	SSDP	374	NOTIFY * HTTP/1.1
10	0.921556	192.168.1.2	239.255.255.250	SSDP	368	NOTIFY * HTTP/1.1
11	0.926079	192.168.1.210	216.58.212.46	TCP	55	50828 → 443 [ACK] Seq=1 Ack=1 Win=61 Len=1 [TCP segment of a reassembled PDU]
12	0.972233	216.58.212.46	192.168.1.210	TCP	66	443 → 50828 [ACK] Seq=1 Ack=2 Win=369 Len=0 SLE=1 SRE=2
13	1.033432	192.168.1.2	239.255.255.250	SSDP	315	NOTIFY * HTTP/1.1
14	1.126303	192.168.1.2	239.255.255.250	SSDP	370	NOTIFY * HTTP/1.1
15	1.228761	192.168.1.2	239.255.255.250	SSDP	380	NOTIFY * HTTP/1.1
16	1.371312	192.168.1.210	176.56.156.22	TCP	54	50830 → 443 [FIN, ACK] Seq=1 Ack=1 Win=64 Len=0
17	1.371445	192.168.1.210	176.56.156.22	TCP	54	50829 → 443 [FIN, ACK] Seq=1 Ack=1 Win=64 Len=0
18	1.371941	192.168.1.210	192.168.1.1	DNS	82	Standard query 0xb78f A ebanking.bankmellat.ir
19	1.372835	192.168.1.210	192.168.1.255	NBNS	92	Name query NB WPAD(00)
20	1.373120	fe80::d804:9e36:eee::ff02::1:3	LUMNR	84	Standard query 0xb10e A wpad	
21	1.373226	192.168.1.210	224.0.0.252	LUMNR	64	Standard query 0xb10e A wpad
22	1.383155	176.56.156.22	192.168.1.210	TCP	60	443 → 50830 [FIN, ACK] Seq=1 Ack=2 Win=245 Len=0
23	1.383205	192.168.1.210	176.56.156.22	TCP	54	50830 → 443 [ACK] Seq=1 Ack=2 Win=64 Len=0

> Frame 1: 306 bytes on wire (2448 bits), 306 bytes captured (2448 bits) on interface DeviceVNF (BB26C44)

Ethernet II, Src: TpLinkTechno\_d6:32:ac (10:fa:ed:d6:32:ac), Dst: IPv4mcast\_7f:ffa (01:00:5e:7f:ff:fa)

Internet Protocol Version 4, Src: 192.168.1.2, Dst: 239.255.255.250

User Datagram Protocol, Src Port: 34718, Dst Port: 1900

Simple Service Discovery Protocol

0000

01 00 5e 7f ff fa 10 fe

ed d6 32 ac 08 00 45 00

.....2...E-

0010

01 24 00 00 40 00 04 11

c4 24 c0 a8 01 02 ef ff

.\$.....\$.....

0020

ff fa 87 9e 07 6c 01 10

cf 67 4e 4f 54 49 46 59

.....1...NOTIFY

0030

20 2a 20 48 54 50 2f 31

2e 31 0d 0a 48 4f 53

\* HTTP/1.1..HOS

0040

54 3a 20 32 33 39 2e 32

35 35 2e 32 35 35 2e 32

T: 239.2.55.255.2

0050

35 30 3a 31 39 30 30 0d

0a 43 41 43 48 45 2d 43

50:1900..CACHE-C

0060

4f 4e 54 52 4f 4c 3a 20

6d 61 78 2d 61 67 65 3d

ONTROL: max-age=

0070

31 30 30 0d 0a 4c 4f 43

41 54 49 4f 4e 3a 20 68

100..LOC ATION: h

0080

74 74 70 3a 2f 2f 31 39

32 2e 31 36 38 2e 31 2e

http://19.2.168.1.

0090

32 3a 31 39 30 30 2f 69

67 64 2e 78 6d 6c 0d 0a

2:1900/1.gd.xml..

00a0

4e 54 3a 20 75 70 6e 70

3a 72 6f 6f 74 64 65 76

NT: upnp:rootdev

00b0

69 63 65 0d 0a 4e 54 53

3a 20 73 73 64 70 3a 61

ice..NTS: ssdp:a

00c0

6c 69 76 65 0d 0a 53 45

52 50 45 52 3a 20 69 70

live..3C RMFES: ap

00d0

0f 73 2f 57 2e 30 20 55

50 6e 50 2f 31 7a 38 20

00/7a:0 PnP/1.0

00e0

54 4c 2d 57 52 3f 34 30

4e 2f 34 2e 30 0d 0a 55

TL..M740 N/4.0..U

00f0

53 4e 3a 20 75 75 69 64

3a 30 36 30 62 37 33 35

SH: uuid:060b735

0100

33 2d 66 63 61 36 2d 34

30 37 30 2d 38 35 66 34

3-fca6-4 070-85f4

catcha.pcapng

Packets: 342 - Displayed: 342 (100.0%)

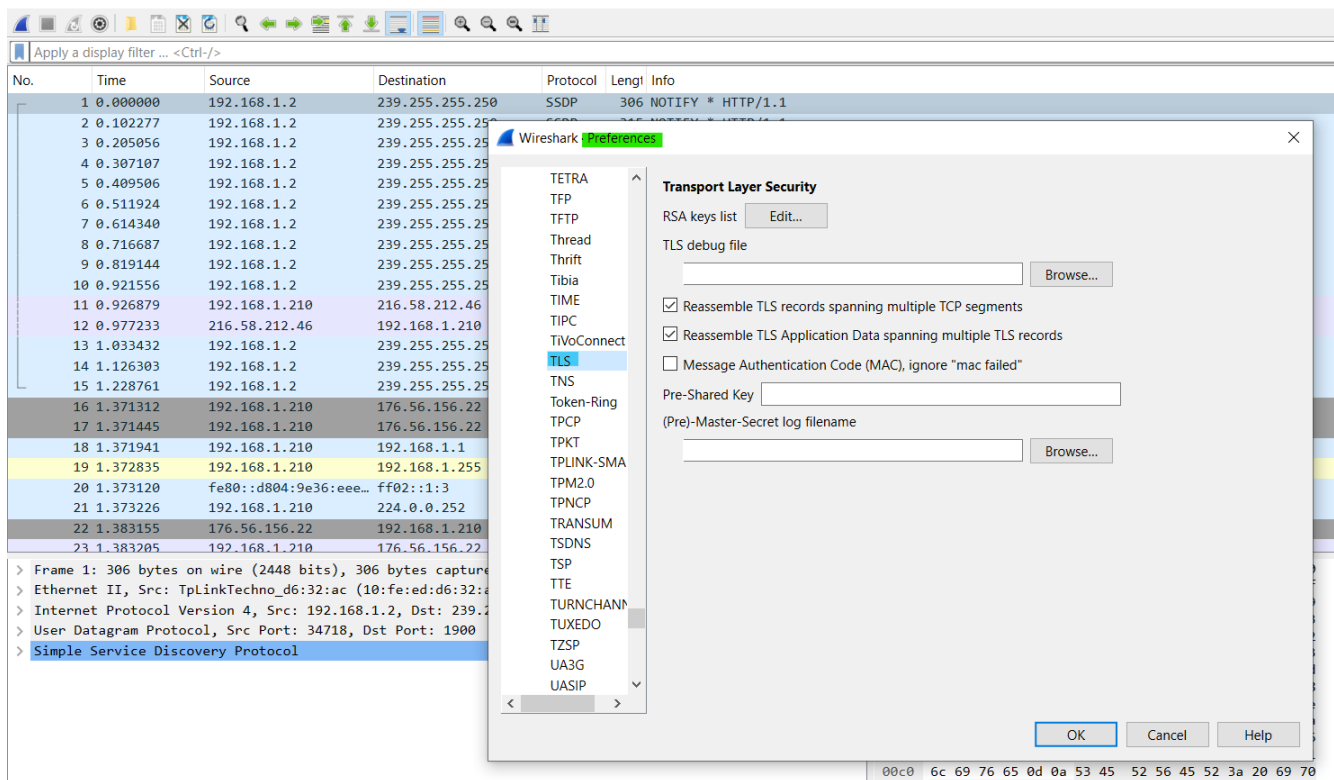
Profile: Default

تصویر 2. فایل captcha.pcapng

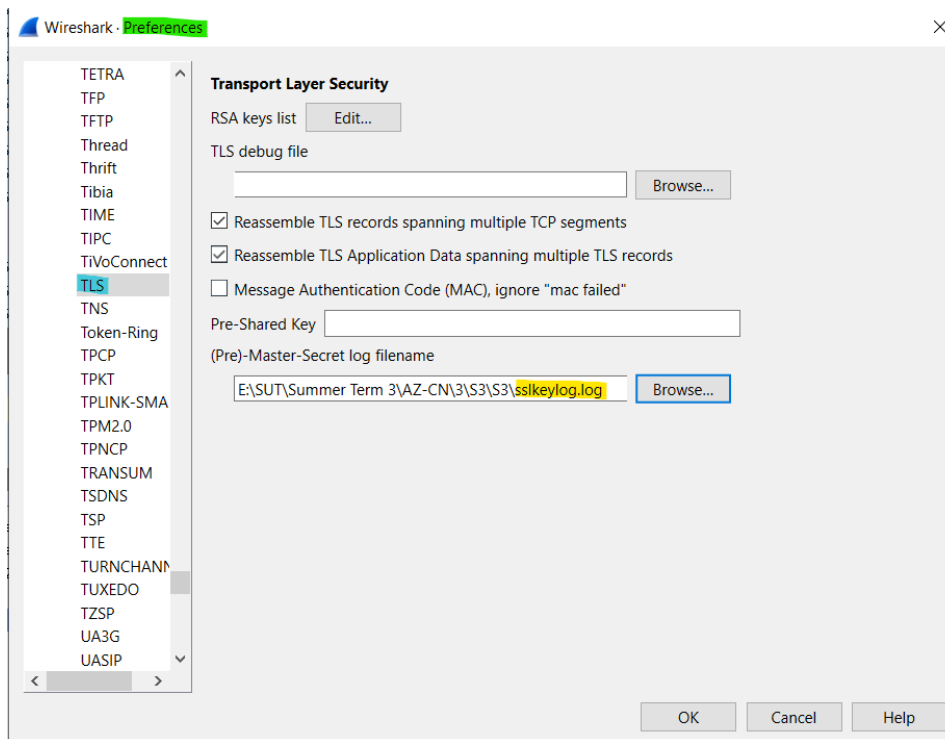
بنابراین تا به اینجا، مرحله اول را با موفقیت انجام داده‌ایم.

حال مطابق دستور کار باید کلید جلسه‌ی مربوط به این فایل ضبط شده از فعالیت‌های شبکه را وارد نرم‌افزار کنیم. بنابراین به قسمت Edit می‌رویم. سپس Preferences را انتخاب می‌کنیم. بعد از آن، به قسمت Protocols می‌رویم. دقت کنید به این دلیل که ssl عملاً deprecated شده است، در نسخه‌های جدید نرم‌افزار وایرشارک، باید از گزینه tls استفاده کنیم (تصویر 3).

سپس در بخش (Pre)-Master-Secret log filename، فایل لاگ داده شده به اسم sslkeylog.log را قرار می‌دهیم (تصویر 4).



تصویر 3. انجام مراحل گفته شده و انتخاب tls



تصویر 4. قرار دادن فایل sslkeylog.log در بخش گفته شده

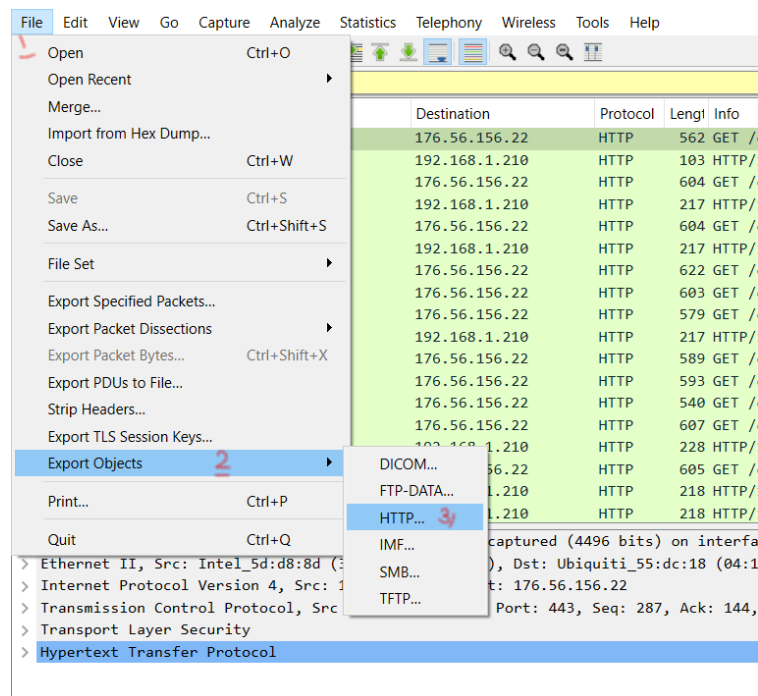
بنابراین بخش دوم را هم با موفقیت انجام دادیم.

برای بخش بعدی به کمک فیلتر `http && ssl`، بسته‌ها را به شکلی فیلتر می‌کنیم که بسته‌های رمزگشایی شده را به ما نشان بدهد. برای فیلتر کردن از نوار بالای صفحه نرم‌افزار کمک می‌گیریم.

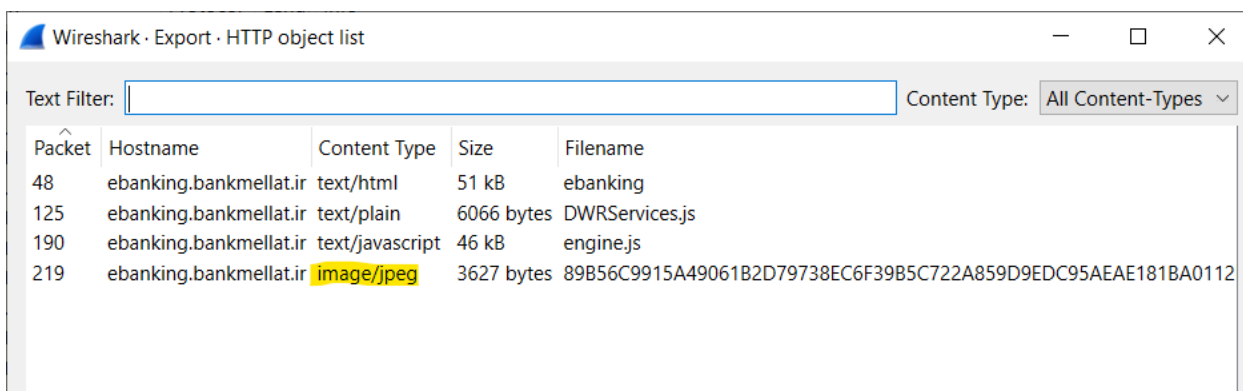
http && ssl						
No.	Time	Source	Destination	Protocol	Length	Info
36	1.571408	192.168.1.210	176.56.156.22	HTTP	562	GET /ebanking/ HTTP/1.1
48	1.607570	176.56.156.22	192.168.1.210	HTTP	103	HTTP/1.1 200 OK (text/html)
50	1.610282	192.168.1.210	176.56.156.22	HTTP	604	GET /ebanking/scripts/VirtualKeyboard/virtualkeyboard.js HTTP/1.1
56	1.620824	176.56.156.22	192.168.1.210	HTTP	217	HTTP/1.1 304 OK
57	1.621960	192.168.1.210	176.56.156.22	HTTP	604	GET /ebanking/styles/ebanking_generic.css HTTP/1.1
73	1.631375	176.56.156.22	192.168.1.210	HTTP	217	HTTP/1.1 304 OK
74	1.632085	192.168.1.210	176.56.156.22	HTTP	622	GET /ebanking/scripts/jquery/jquery-ui-1.11.4/jquery-ui.css HTTP/1.1
81	1.634372	192.168.1.210	176.56.156.22	HTTP	603	GET /ebanking/styles/virtualKeyboard.css HTTP/1.1
89	1.643090	192.168.1.210	176.56.156.22	HTTP	579	GET /ebanking/scripts/expand.js HTTP/1.1
107	1.646663	176.56.156.22	192.168.1.210	HTTP	217	HTTP/1.1 304 OK
108	1.647375	192.168.1.210	176.56.156.22	HTTP	589	GET /ebanking/scripts/ebanking_generic.js HTTP/1.1
110	1.647537	192.168.1.210	176.56.156.22	HTTP	593	GET /ebanking/scripts/jquery/jquery-1.11.3.js HTTP/1.1
111	1.647843	192.168.1.210	176.56.156.22	HTTP	540	GET /ebanking/dwr/interface/DWRServices.js HTTP/1.1
112	1.647969	192.168.1.210	176.56.156.22	HTTP	607	GET /ebanking/dwr/engine.js HTTP/1.1
113	1.650300	176.56.156.22	192.168.1.210	HTTP	228	HTTP/1.1 304 Not Modified
114	1.650866	192.168.1.210	176.56.156.22	HTTP	605	GET /ebanking/dwr/util.js HTTP/1.1
116	1.654181	176.56.156.22	192.168.1.210	HTTP	218	HTTP/1.1 304 OK
120	1.664668	176.56.156.22	192.168.1.210	HTTP	218	HTTP/1.1 304 OK
121	1.668977	176.56.156.22	192.168.1.210	HTTP	228	HTTP/1.1 304 Not Modified
122	1.668977	176.56.156.22	192.168.1.210	HTTP	204	HTTP/1.1 304 Not Modified
125	1.676407	176.56.156.22	192.168.1.210	HTTP	103	HTTP/1.1 200 OK (text/plain)
166	1.710158	192.168.1.210	176.56.156.245	HTTP	584	GET /ebanking/images/bg_loginUserIcn.png HTTP/1.1
190	1.729223	176.56.156.22	192.168.1.210	TLSv1.2	221	HTTP/1.1 200 OK (text/javascript)
191	1.730312	176.56.156.245	192.168.1.210	HTTP	232	HTTP/1.1 304 Not Modified
192	1.731160	192.168.1.210	176.56.156.245	HTTP	584	GET /ebanking/images/bg_loginPwdIcn.png HTTP/1.1
193	1.732641	192.168.1.210	176.56.156.22	HTTP	641	GET /ebanking/PassImageServlet/89B56C9915A49061B2D79738EC6F39B5C722A859D9 HTTP/1.1
199	1.745267	176.56.156.245	192.168.1.210	HTTP	231	HTTP/1.1 304 Not Modified
215	1.751638	192.168.1.210	176.56.156.245	HTTP	585	GET /ebanking/images/bg_main_right01.png HTTP/1.1
216	1.752360	176.56.156.22	192.168.1.210	HTTP	88	HTTP/1.1 200 OK (image)

تصویر 5. فیلتر `http && ssl`

حال باید تصویر داده شده در این ارتباط را ذخیره نماییم. برای این کار، به بخش `File` رفته و زیر بخش `Export` `objects` را انتخاب می‌کنیم. سپس `HTTP` را انتخاب کرده تا آبجکت‌هایی که تحت این پروتکل قابل خروجی گرفتن هستند را ببینیم.

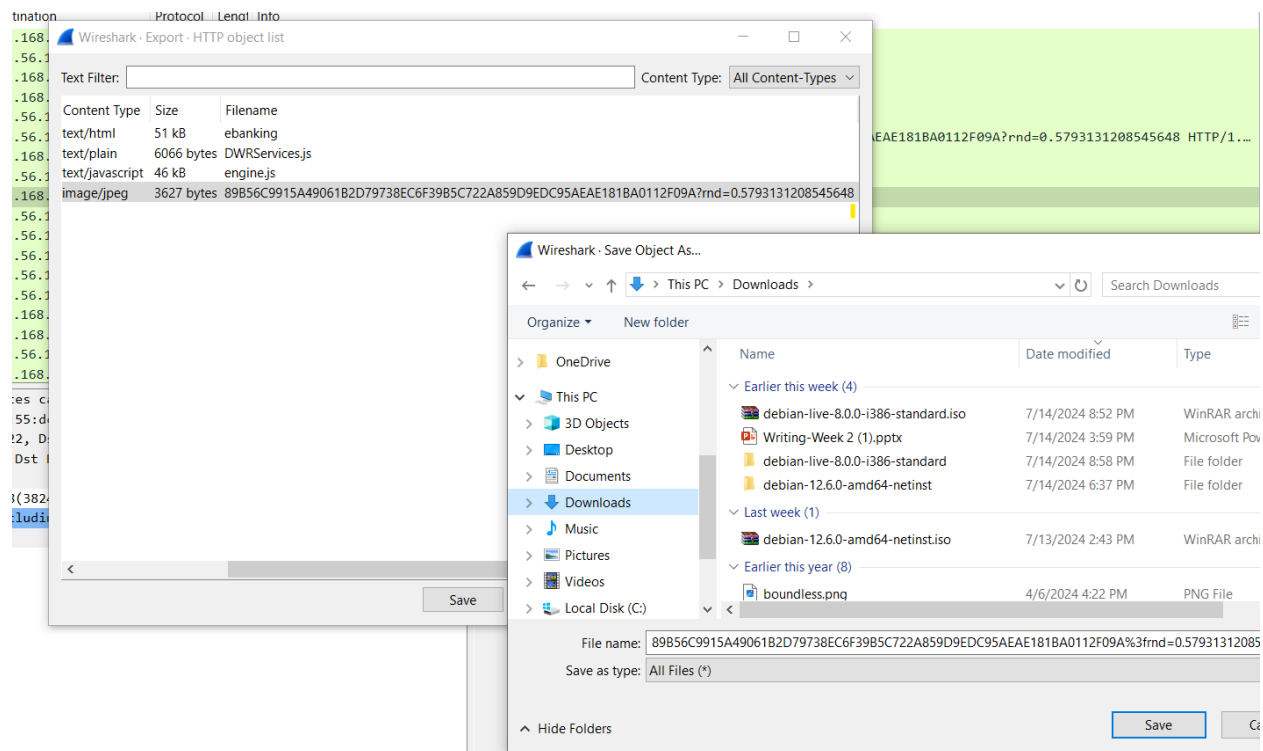


تصویر 6. نحوه به دست آوردن objectها



تصویر 7. Objectهای موجود خواسته شده

همان طور که در تصویر 7 مشاهده می کنید، یک فایل html، دو فایل java script و یک jpeg مشاهده می کنید که این همان تصویری است که دنبال آن هستیم. تصویر را انتخاب کرده، گزینه save را زده و در کامپیوتر خود ذخیره می نماییم.



تصویر 8. مراحل ذخیره کردن تصویر



تصویر 9. تصویر کپچا خواسته شده

بنابراین بخش اول این آزمایش به اتمام رسید و تصویر مورد نظر را نشان دادیم.

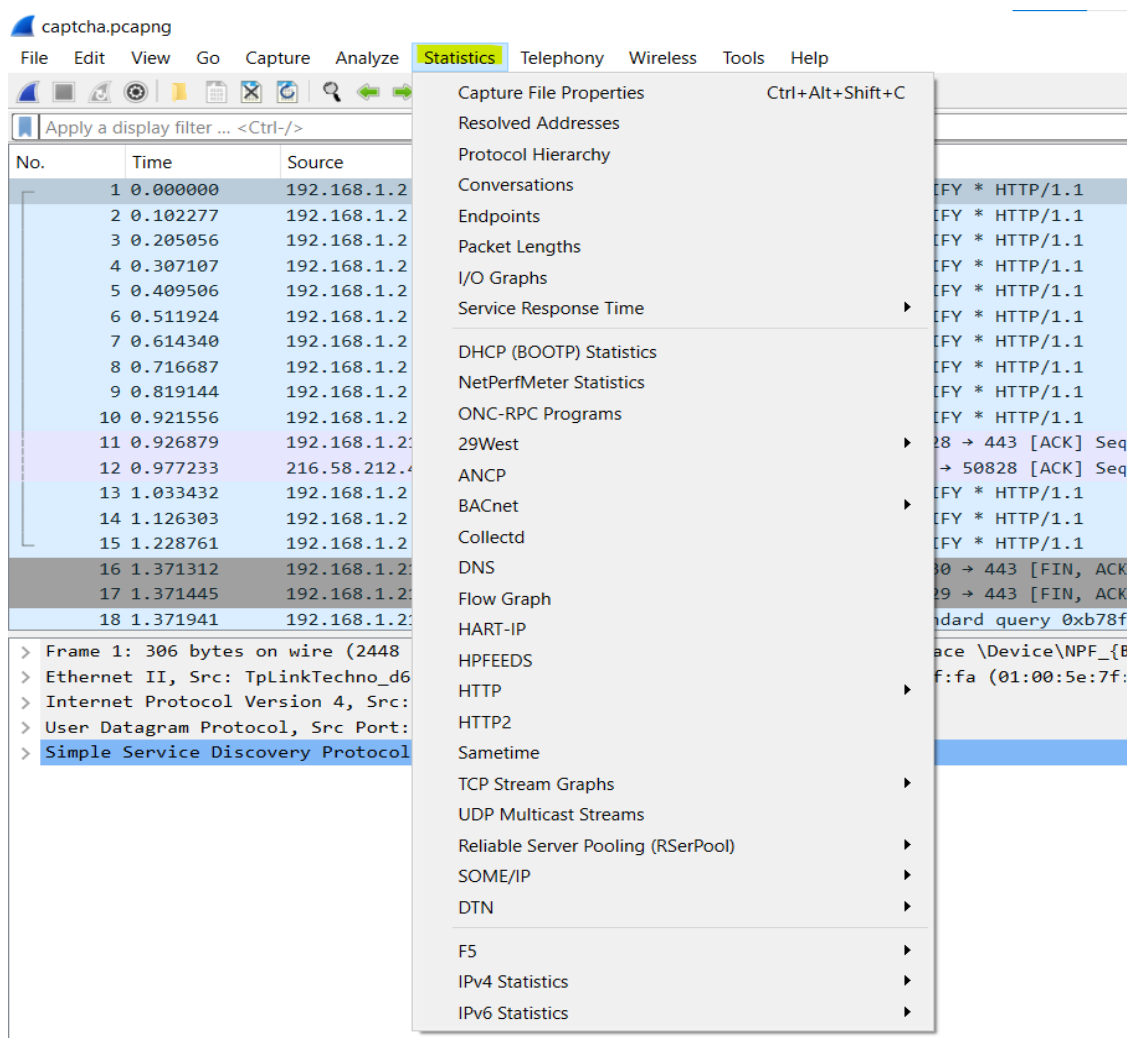


## 1.2. سوال‌ها

1. تحقیق کنید ببینید که اطلاعات آماری بسته‌ها در نرم افزار wireshark به چه صورت به دست می‌آید و بگویید که این اطلاعات به ما چه کمک‌هایی می‌توانند بکنند. به طور مثال یکی از کمک‌های این اطلاعات آماری زمانی است که ما کلید جلسه بسته‌های رمز شده را نداشته باشیم. اگر خواستید می‌توانید در این زمینه بیش‌تر تحقیق کنید.

پاسخ:

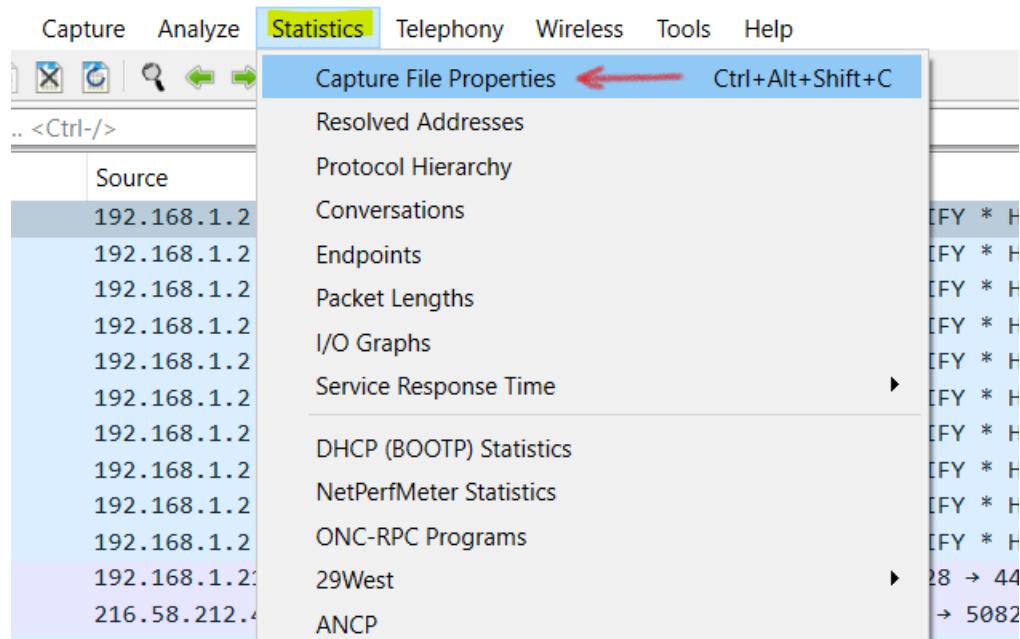
در نرم‌افزار وایرشارک بخشی به نام statistics وجود دارد که اطلاعات و آماره‌های گوناگونی از طریق آن قابل استخراج و دسترسی می‌باشد.



تصویر 10. بخش statistics نرم‌افزار وایرشارک

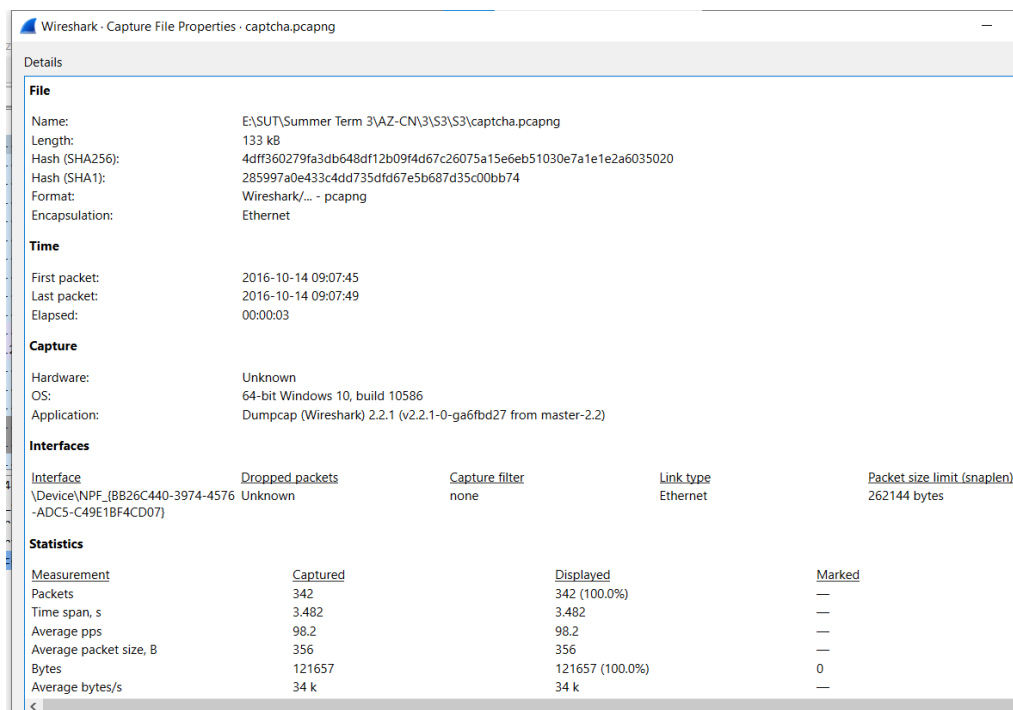
حال چند زیر بخش آن را توضیح خواهیم داد.

## بخش Capture file properties



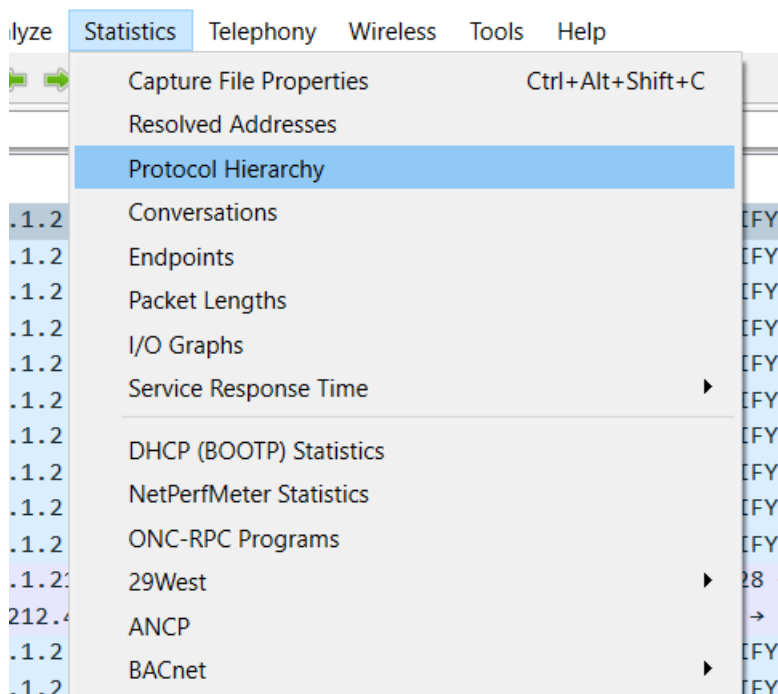
تصویر 11. بخش capture file properties

این بخش اطلاعات کلی فایل ضبط شده را دارا است. زمان شروع و پایان، interface ی که از آن این اطلاعات ضبط شده است و تعداد پکت‌های کپچر شده از جمله این اطلاعات هستند.



تصویر 12. بخش capture file properties

همانطور مشاهده می‌کنیم، در این بسته فرمت کدگذاری فایل مشخص شده که به واسطه آن شاید بتوان برخی فایل‌ها را با تکنیک‌های رمزنگاری، رمزگشایی کرد.  
زیربخش بعدی، protocol hierarchy می‌باشد.



تصویر 13. بخش protocol hierarchy

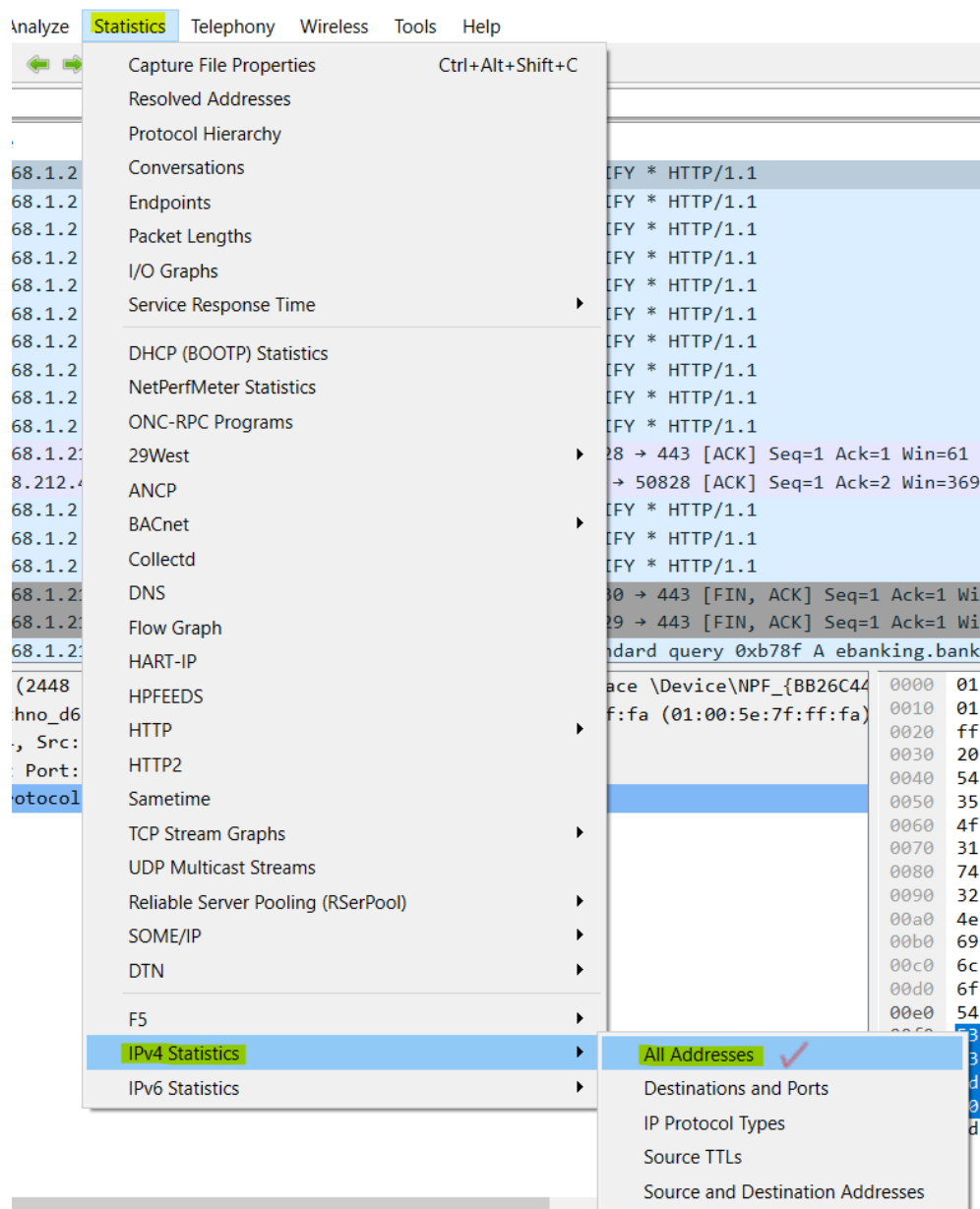
این بخش، شامل یک دسته‌بندی درختی از پروتکل‌ها و اطلاعات مهمی مانند سهم هر پروتکل از تعداد پکت است. تصویر این بخش را در تصویر 14 مشاهده می‌نمایید:

Wireshark - Protocol Hierarchy Statistics - captcha.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDU's
▼ Frame	100.0	342	100.0	121657	279 k	0	0	0	342
▼ Ethernet	100.0	342	4.1	4938	11 k	0	0	0	342
▼ Internet Protocol Version 6	0.6	2	0.1	80	183	0	0	0	2
▼ User Datagram Protocol	0.6	2	0.0	16	36	0	0	0	2
Link-local Multicast Name Resolution	0.6	2	0.0	44	101	2	44	101	2
▼ Internet Protocol Version 4	99.4	340	5.6	6800	15 k	0	0	0	340
▼ User Datagram Protocol	10.2	35	0.2	280	643	0	0	0	35
Simple Service Discovery Protocol	3.8	13	3.3	4000	9190	13	4000	9190	13
NetBIOS Name Service	0.9	3	0.1	150	344	3	150	344	3
Link-local Multicast Name Resolution	0.6	2	0.0	44	101	2	44	101	2
Domain Name System	5.0	17	0.9	1130	2596	17	1130	2596	17
▼ Transmission Control Protocol	89.2	305	85.6	104175	239 k	147	57961	133 k	305
▼ Transport Layer Security	46.2	158	80.6	98008	225 k	68	36988	84 k	160
▼ Hypertext Transfer Protocol	26.3	90	73.8	89727	206 k	86	28684	65 k	90
Line-based text data	0.9	3	84.9	103343	237 k	3	103343	237 k	3
JPEG File Interchange Format	0.3	1	3.0	3627	8333	1	3627	8333	1

تصویر 14. بخش protocol hierarchy

همچنین بخش IPv4 statistics نیز دارای چند زیر بخش است که زیر بخش All Addresses شامل تمام آی‌پی‌های IPv4 که مبدا و مقصد پکت‌های ضبط شده بودند و تعداد بسته‌های مرتبط با آن‌ها و اطلاعات دیگر را دید.



تصویر 15. بخش IPv4 Statistics و All Addresses

و اطلاعات زیر مشاهده می‌شود:

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
▼ All Addresses	340				0.0977	100%	1.3700	1.701
239.255.255.250	13				0.0037	3.82%	0.0200	1.033
224.0.0.252	2				0.0006	0.59%	0.0100	1.373
216.58.212.46	2				0.0006	0.59%	0.0200	0.927
192.168.1.255	3				0.0009	0.88%	0.0100	1.373
192.168.1.210	327				0.0939	96.18%	1.3700	1.701
192.168.1.2	13				0.0037	3.82%	0.0200	1.033
192.168.1.1	17				0.0049	5.00%	0.0800	1.819
176.56.156.245	138				0.0396	40.59%	1.1300	1.730
176.56.156.22	165				0.0474	48.53%	1.1000	1.621

تصویر 16. بخش All Addresses

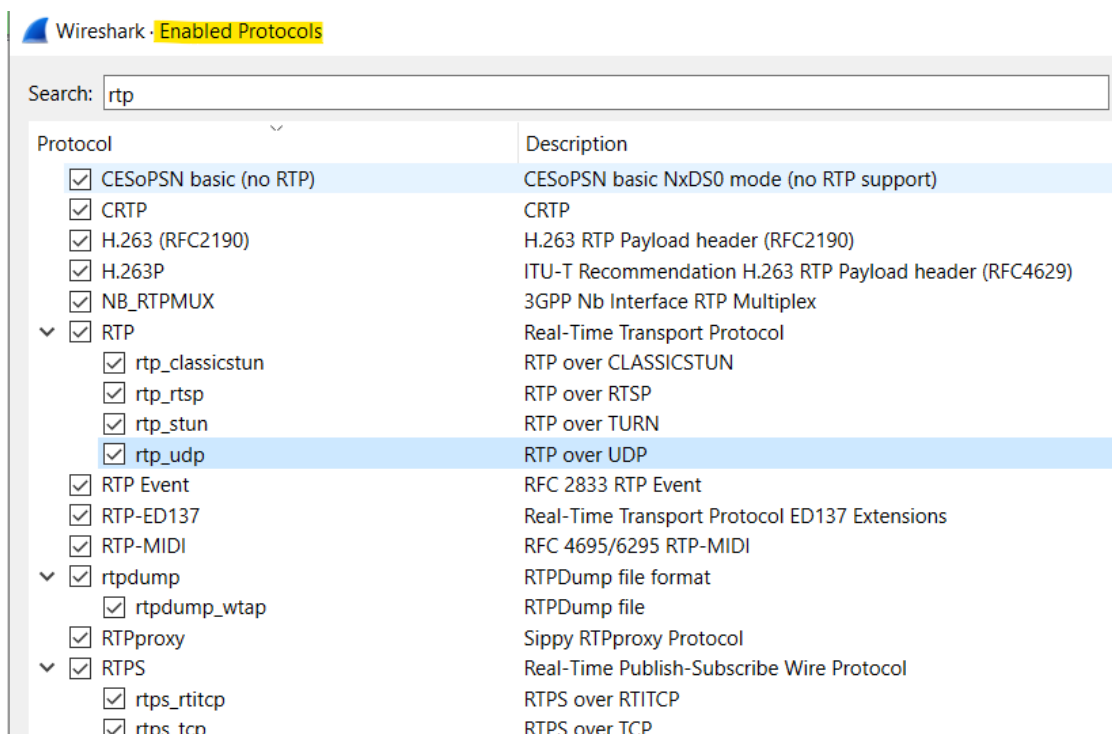
به طور کلی بخش statistics، بخش بسیار مهمی در نرم‌افزار وایرشارک به حساب می‌آید و اطلاعات مختلف برای فایل pcapng مانند زمان، تعداد و سرعت و حجم و ... برای پکت‌ها، نمایش اتصالات برقرار شده با تفکیک IP و پورت مبدا و مقصد با ارائه آماری، اطلاعات آماری طول پکت‌ها و ... می‌باشد.

## 2. پروتکل RTP چیست و توضیح دهید که چگونه نرم‌افزار wireshark در تحلیل آن به ما کمک می‌کند.

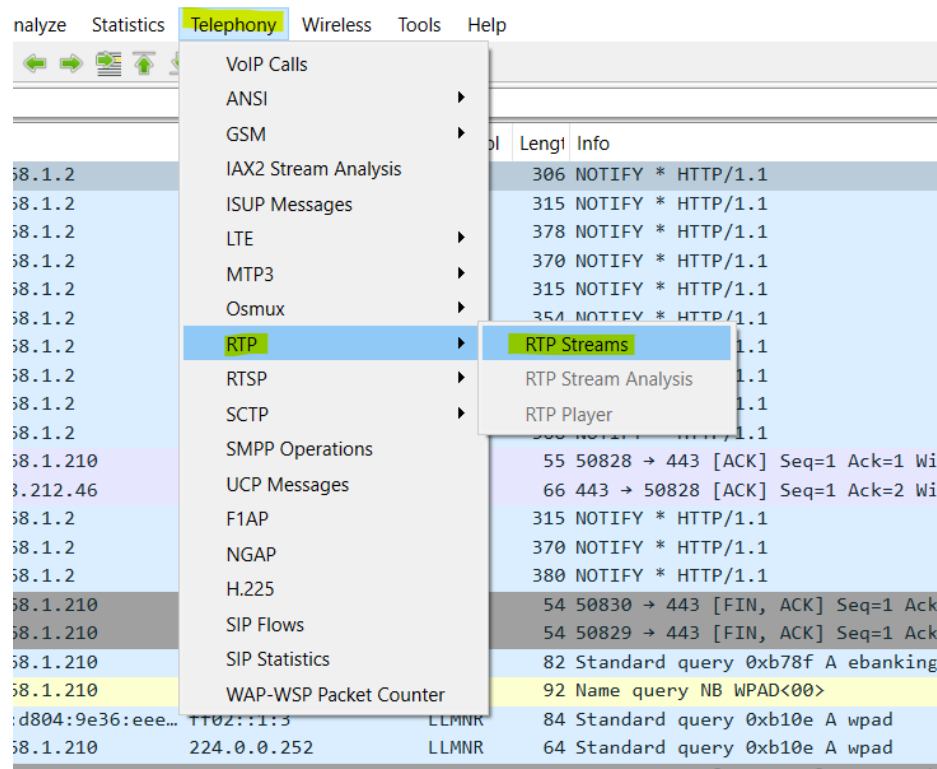
پاسخ:

پروتکل Real-Time Transfer Protocol یا همان RTP، یک پروتکل برای انتقال محتوای چندرسانه‌ای مانند تصویر و صدا تحت شبکه‌های آی‌پی می‌باشد. این پروتکل معمولاً با یک پروتکل سیگنال مانند SIP استفاده می‌شود که به منزله شروع استریم داده است. این پروتکل برای انتقال محتوا در استفاده‌های ارتباطاتی و سرگرمی که شامل انتقال صدا و تصاویر و ویدئوهاست استفاده می‌شود. بسته‌های این پروتکل از طریق پروتکل UDP در لایه‌ی انتقال، منتقل می‌شوند. همچنین بسته‌های دیگری تحت عنوان RTCP تحت این پروتکل ارسال می‌شوند که به انتقال آماره‌های مربوط به انتقال محتوای چندرسانه‌ای می‌پردازند. در نرم‌افزار وایرشارک، از 2 طریق می‌توان بسته‌های rtp ارتباط داشت.

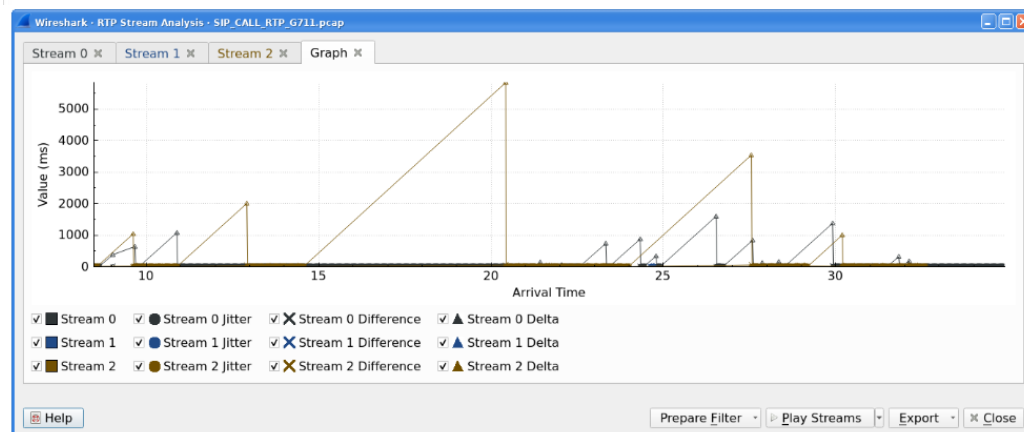
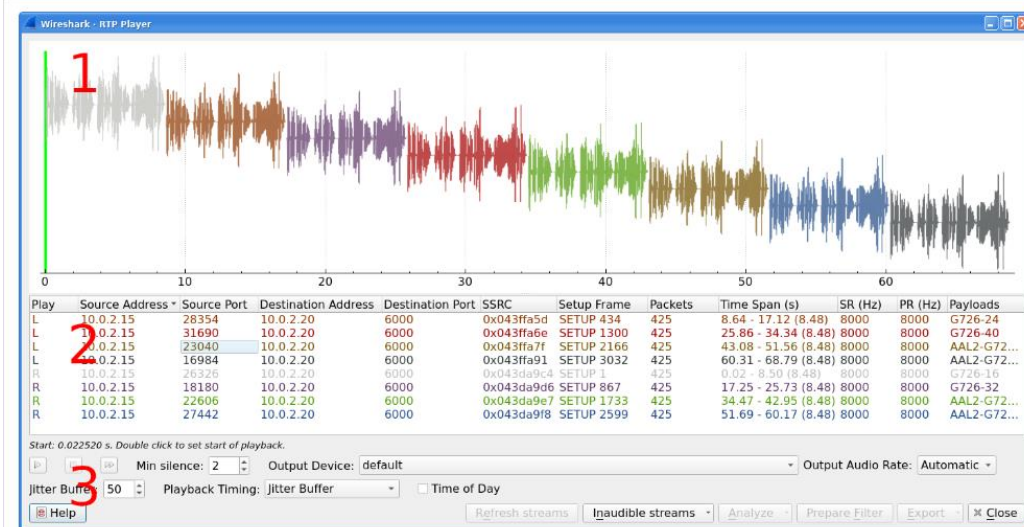
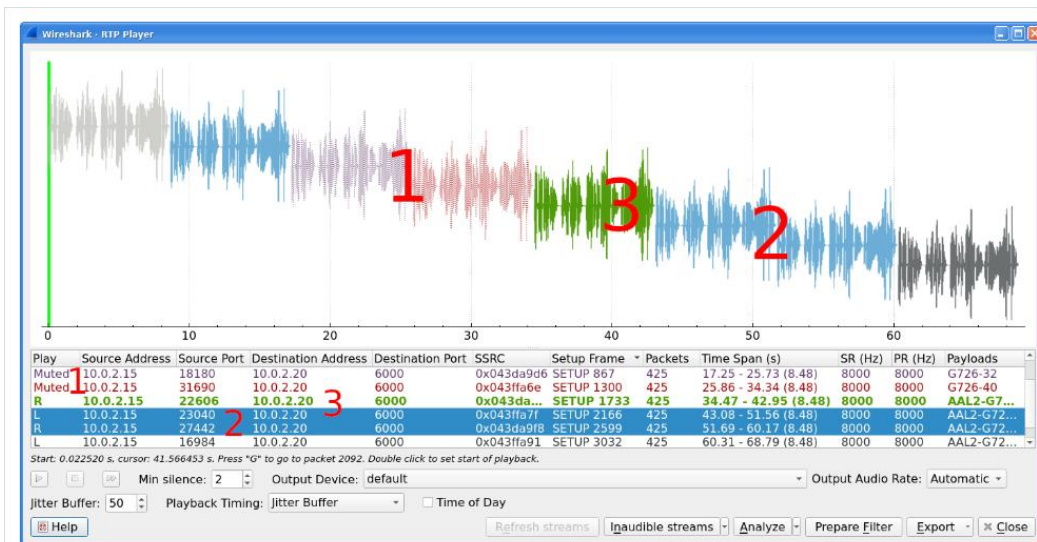
با فیلتر کردن پکت‌ها با فیلتر rtp می‌توان بسته‌های منتقل شده تحت این پروتکل را پیدا کرد. همچنین از بخش telephony و زیربخش rtp، می‌توان ابزارهای مفیدی برای تحلیل ارتباطات تحت این پروتکل پیدا کرد. البته توجه کنید، در صورتی که بسته با رمزگذاری tls رمزگذاری شده باشد، وایرشارک نمی‌تواند آن را به عنوان rtp تشخیص دهد. برای اطلاعات بیشتر به این [لینک](#) که وسایت وایرشارک و بخش مربوط به rtp است، می‌توانید مراجعه کنید.



تصویر 17. بخش Analyze و Enabled Protocols و فعال سازی RTP



تصویر 18. بخش Telephony و زیربخش RTP



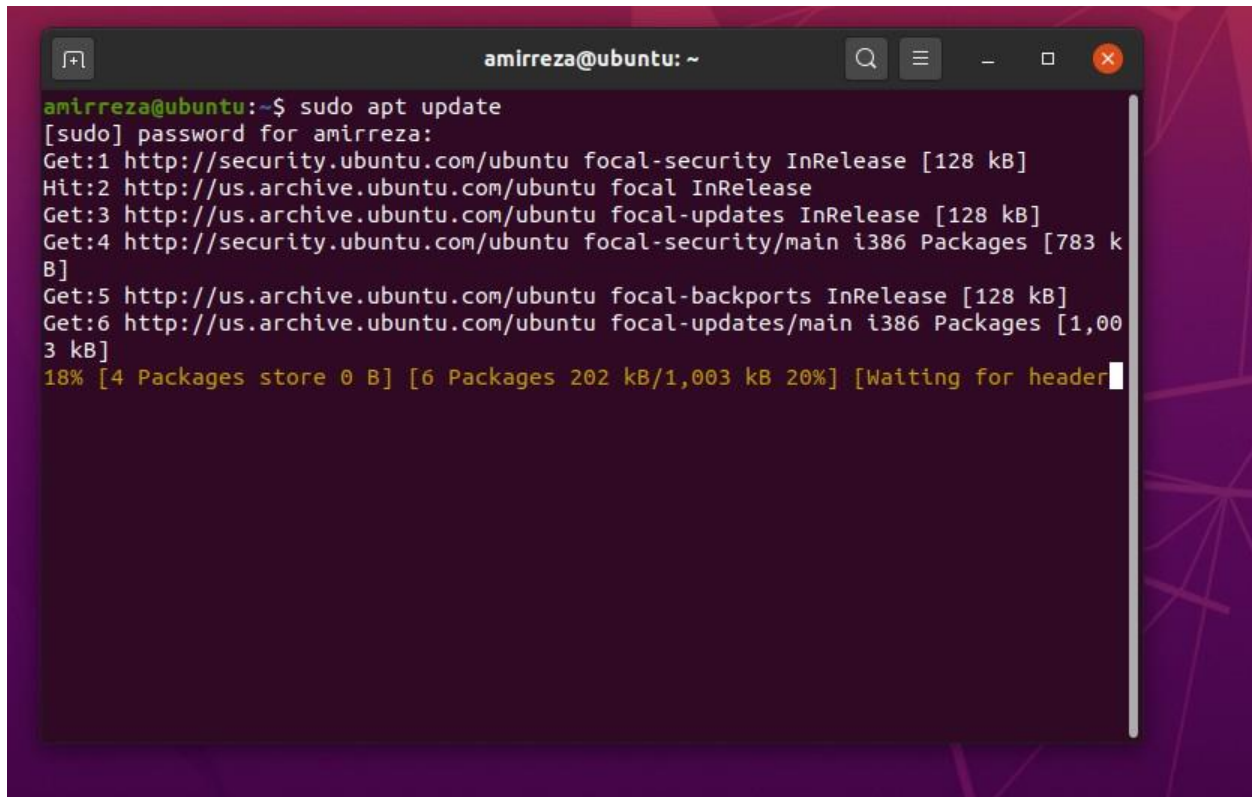
تصویر 19. تصاویر مربوط به rtp در سایت وایرشارک



## بخش دوم \_ راه اندازی DNS

### 2.1. سناریو آزمایش

در این بخش ابتدا از ubuntu استفاده کردم. اما به دلیل برخوردن به ارور نامشخص، از Debian 8 استفاده کردم. ابتدا مراحل نصب bind9 را می بینیم:



```
amirreza@ubuntu:~$ sudo apt update
[sudo] password for amirreza:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [783 k
B]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [1,00
3 kB]
18% [4 Packages store 0 B] [6 Packages 202 kB/1,003 kB 20%] [Waiting for header
```

تصویر 20. Sudo apt update



```
amirreza@ubuntu:~$ sudo apt install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bind9-dnsutils bind9-libs bind9-utils python3-ply
Suggested packages:
  bind-doc resolvconf python-ply-doc
The following NEW packages will be installed:
  bind9 bind9-utils python3-ply
The following packages will be upgraded:
  bind9-dnsutils bind9-libs
2 upgraded, 3 newly installed, 0 to remove and 422 not upgraded.
Need to get 1,734 kB of archives.
After this operation, 2,067 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

تصویر 21. Sudo apt install bind9



```

amirreza@ubuntu:~$ nslookup google.com 127.0.0.1
Server:          127.0.0.1
Address:         127.0.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 216.239.38.120
Name:   google.com
Address: 2001:4860:4802:32::78

amirreza@ubuntu:~$

```

تصویر 22. چک کردن صحت نصب bind9

حال به بخش `/etc/bind/named.conf.options` رفته و forwarder قرار می‌دهیم که برای این کار از گوگل استفاده می‌کنیم.

```

GNU nano 4.8 /etc/bind/named.conf.options Modified
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; };
};

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo

```

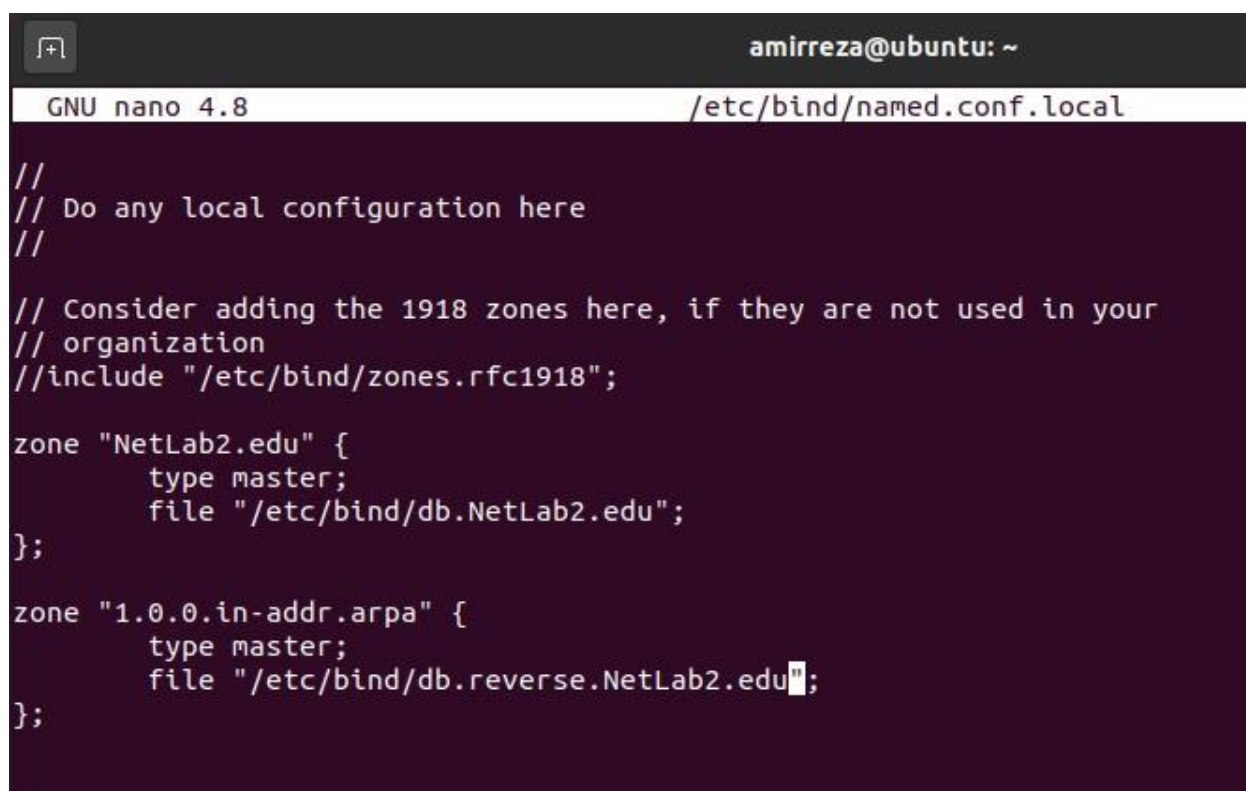
تصویر 23. اضافه کردن forwarder

سپس درستی را چک کرده و bind9 را ری‌استارت می‌کنیم تا تغییرات صورت بگیرد.

```
amirreza@ubuntu:~$ sudo nano /etc/bind/named.conf.options
amirreza@ubuntu:~$ sudo named-checkconf
amirreza@ubuntu:~$ sudo systemctl restart bind9
amirreza@ubuntu:~$
```

تصویر 24. ری‌استارت کردن bind9

در مرحله بعد به سراغ `/etc/bind/named.conf.local` رفته و به شکل زیر تغییرش می‌دهیم.



```
amirreza@ubuntu: ~
GNU nano 4.8 /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "NetLab2.edu" {
    type master;
    file "/etc/bind/db.NetLab2.edu";
};

zone "1.0.0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.reverse.NetLab2.edu";
};
```

تصویر 25. مشخص کردن zone ها و اطلاعات آن‌ها

```

amirreza@ubuntu:~$ sudo nano /etc/bind/named.conf.local
amirreza@ubuntu:~$ cat /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "NetLab2.edu" {
    type master;
    file "/etc/bind/db.NetLab2.edu";
};

zone "1.0.0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.reverse.NetLab2.edu";
};

amirreza@ubuntu:~$

```

تصویر 26. خروجی فایل `/etc/bind/named.conf.local`

Zone اول مطابق دستور کار NetLab2.edu قرار داده شده و از نوع master است. آدرس فایل دیتابیس نیز در خط بعد از آن تایپ شده است. این منطقه برای جواب دادن درخواست‌های عادی dns استفاده خواهد شد.

Zone دوم از نوع reserved dns است. این منطقه نیز از نوع master است و آدرس فایل دیتابیس آن نیز در خط بعد مشخص شده است. این منطقه برای جواب دادن درخواست‌های عادی rdns استفاده خواهد شد.

حال باید فایل‌های دیتابیس مشخص نموده را پر کنیم. از اینجا به بعد از Debian کمک گرفتیم.

```
root@debian:/etc/bind# cat /etc/bind/db.NetLab2.edu
$TTL      604800
@         IN      SOA      ns.NetLab2.edu. admin.NetLab2.edu. (
                        8      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TT;
@         IN      NS       ns.NetLab2.edu.
ns        IN      A        127.0.0.1

g1        IN      CNAME    group1
group1    IN      A        127.0.0.2

g2        IN      CNAME    group2
group2    IN      A        127.0.0.3
root@debian:/etc/bind#
```

تصویر 27. خروجی فایل db.NetLab2.edu

```
root@debian:/etc/bind# cat /etc/bind/db.NetLab2.edu
$TTL      604800
@         IN      SOA      ns.NetLab2.edu. admin.NetLab2.edu. (
                        8      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TT;
@         IN      NS       ns.NetLab2.edu.
ns        IN      A        127.0.0.1

g1        IN      CNAME    group1
group1    IN      A        127.0.0.2

g2        IN      CNAME    group2
group2    IN      A        127.0.0.3
root@debian:/etc/bind# cat /etc/bind/db.reverse.NetLab2.edu
$TTL      604800

@         IN      SOA      ns.NetLab2.edu. admin.NetLab2.edu. (
                        3      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;

@         IN      NS       ns.NetLab2.edu.

1.0.0     IN      PTR      ns.NetLab2.edu.
2.0.0     IN      PTR      group1.NetLab2.edu.
3.0.0     IN      PTR      group2.NetLab2.edu.
root@debian:/etc/bind# _
```

تصویر 28. خروجی دو فایل گفته شده

حال مراحل زیر را اجرا می‌کنیم:

```
root@debian:/etc/bind# sudo named-checkzone 1.0.0.in-addr.arpa db.reverse.NetLab2.edu
zone 1.0.0.in-addr.arpa/IN: loaded serial 3
OK
root@debian:/etc/bind# sudo named-checkzone NetLab2.edu db.NetLab2.edu
zone NetLab2.edu/IN: loaded serial 8
OK
root@debian:/etc/bind# _
```

تصویر 29. انجام دستورات named-checkzone و موفقیت

```
root@debian:/etc/bind# sudo service bind9 restart
root@debian:/etc/bind# service bind9 status
• bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled)
   Drop-In: /run/systemd/generator/bind9.service.d
            └─50-insserv.conf-$named.conf
   Active: active (running) since Fri 2024-07-19 13:52:33 EDT; 6s ago
     Docs: man:named(8)
  Process: 2392 ExecStop=/usr/sbin/rndc stop (code=exited, status=0/SUCCESS)
    Main PID: 2397 (named)
      CGroup: /system.slice/bind9.service
              └─2397 /usr/sbin/named -f -u bind

Jul 19 13:52:34 debian named[2397]: command channel listening on ::1#953
Jul 19 13:52:34 debian named[2397]: managed-keys-zone: loaded serial 2
Jul 19 13:52:34 debian named[2397]: zone 0.in-addr.arpa/IN: loaded serial 1
Jul 19 13:52:34 debian named[2397]: zone 1.0.0.in-addr.arpa/IN: loaded serial 3
Jul 19 13:52:34 debian named[2397]: zone 127.in-addr.arpa/IN: loaded serial 1
Jul 19 13:52:34 debian named[2397]: zone localhost/IN: loaded serial 2
Jul 19 13:52:34 debian named[2397]: zone 255.in-addr.arpa/IN: loaded serial 1
Jul 19 13:52:34 debian named[2397]: zone NetLab2.edu/IN: loaded serial 8
Jul 19 13:52:34 debian named[2397]: all zones loaded
Jul 19 13:52:34 debian named[2397]: running
root@debian:/etc/bind# _
```

تصویر 30. ری‌استارت کردن و وضعیت bind9

```
root@debian:/etc/bind# sudo service bind9 restart
root@debian:/etc/bind# nslookup NetLab2.edu
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   NetLab2.edu
Address: 127.0.0.1

root@debian:/etc/bind#
```

تصویر 31. چک کردن درستی عملکرد

```
Address: 127.0.0.1
```

```
root@debian:/etc/bind# host ns.NetLab2.edu
ns.NetLab2.edu has address 127.0.0.1
root@debian:/etc/bind# host group1.NetLab2.edu
group1.NetLab2.edu has address 127.0.0.2
^[[Aroot@debian:/etc/bind# host group1.NetLab2.edu
group1.NetLab2.edu has address 127.0.0.2
root@debian:/etc/bind# host group2.NetLab2.edu
group2.NetLab2.edu has address 127.0.0.3
root@debian:/etc/bind#
```

تصویر 32. چک کردن درستی عملکرد

```
root@debian:/etc/bind# dig group2.NetLab2.edu

; <<>> DiG 9.9.5-9+deb8u15-Debian <<>> group2.NetLab2.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18115
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;group2.NetLab2.edu.          IN      A

;; ANSWER SECTION:
group2.NetLab2.edu.          604800  IN      A      127.0.0.3

;; AUTHORITY SECTION:
NetLab2.edu.                 604800  IN      NS      ns.NetLab2.edu.

;; ADDITIONAL SECTION:
ns.NetLab2.edu.              604800  IN      A      127.0.0.1

;; Query time: 42 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Jul 19 14:39:56 EDT 2024
;; MSG SIZE rcvd: 96

root@debian:/etc/bind#
```

تصویر 33. دستور dig

```

root@debian:/etc/bind# dig group1.NetLab2.edu

; <<>> DiG 9.9.5-9+deb8u15-Debian <<>> group1.NetLab2.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65351
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;group1.NetLab2.edu.                IN      A

;; ANSWER SECTION:
group1.NetLab2.edu.                604800  IN      A      127.0.0.2

;; AUTHORITY SECTION:
NetLab2.edu.                      604800  IN      NS      ns.NetLab2.edu.

;; ADDITIONAL SECTION:
ns.NetLab2.edu.                   604800  IN      A      127.0.0.1

;; Query time: 291 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Jul 19 14:40:40 EDT 2024
;; MSG SIZE rcvd: 96

root@debian:/etc/bind# _

```

تصویر 34. دستور dig

```

root@debian:/etc/bind# dig google.com

; <<>> DiG 9.9.5-9+deb8u15-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52831
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.                        IN      A

;; ANSWER SECTION:
google.com.                        60      IN      A      216.239.38.120

;; AUTHORITY SECTION:
.                258847  IN      NS      j.root-servers.net.
.                258847  IN      NS      l.root-servers.net.
.                258847  IN      NS      i.root-servers.net.
.                258847  IN      NS      h.root-servers.net.
.                258847  IN      NS      k.root-servers.net.
.                258847  IN      NS      b.root-servers.net.
.                258847  IN      NS      e.root-servers.net.
.                258847  IN      NS      f.root-servers.net.
.                258847  IN      NS      g.root-servers.net.
.                258847  IN      NS      d.root-servers.net.
.                258847  IN      NS      m.root-servers.net.
.                258847  IN      NS      c.root-servers.net.
.                258847  IN      NS      a.root-servers.net.

;; Query time: 596 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Jul 19 14:41:48 EDT 2024
;; MSG SIZE rcvd: 266

root@debian:/etc/bind# _

```

تصویر 35. دستور dig

بنابراین این بخش هم به طور کامل انجام شد.

## 2.2. سوال‌ها

برای بخش اول بعد از راه‌اندازی وایرشارک در حالت any، شروع به ضبط و کیچر بسته‌ها می‌نماییم. یک دستور dig می‌زنیم و با فیلتر (ip.src == 127.0.0.1 || ip.dst == 127.0.0.1) && dns بسته‌ها را فیلتر می‌کنیم. سپس خواهیم دید که dig هم به سرور و هم به سرور اصلی همزمان درخواست می‌زند و در نهایت از سرور ما پاسخ می‌گیرد.

برای بخش دوم نیز اگر همان dig g2.NetLab2.edu را بزنیم، خواهیم دید که دو نوع رکورد A و CNAME که در تصاویر قبلی نشان دادیم، دیده می‌شود.

رکوردهای A رکوردهای آدرس هستند که آدرس مربوط به نام جست و جو شده در آن‌ها می‌باشد.

رکوردهای CNAME نیز مربوط به نام‌های مستعار هستند که به طور مثال و همانطور که در تصاویر قبلی مشخص بود و مشخص کردیم (تصویر 27)، g1.NetLab2.edu یک نام مستعار برای group1.NetLab2.edu بود.



## منابع و مراجع:

- [Real-time Transport Protocol - Wikipedia](#)
- [9.11. RTP](#)
- [Configure BIND9 as a Primary DNS Server on Ubuntu 20.04 | Serverspace](#)