آزمایش شماره ۲

آز شبکه ـ دکتر بردیا صفایی

دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف

نیم‌سال اول ۰۱ـ۰۲

گروه ۰:
مهرشاد میرمحمدی ـ ۹۸۱۰۹۶۳۴
پرهام صارمی ـ ۹۷۱۰۱۹۵۹
محمدرضا مفیضی ـ ۹۸۱۰۶۰۵۹

۱ بخش اول

با توجه به اینکه سایت sharif.edu از HTTPS استفاده می‌کرد از سایت old.sharif.edu استفاده شده است. زیر پس از capture کردن اطلاعات روی دامنه‌ی sharif.edu و فیلتر کردن آن خروجی‌ای مشاهده نشد. خروجی این عملیات را می‌توان در شکل ۱ مشاهده کرد.

http

No.	Time	Source	Destination	Protocol	Length	Info
29…	31.725828263	192.168.43.152	81.31.186.20	HTTP	664	GET / HTTP/1.1
29…	31.847509031	81.31.186.20	192.168.43.152	HTTP	435	HTTP/1.1 200 OK (text/html)
29…	32.147975168	192.168.43.152	81.31.186.20	HTTP	697	GET /c HTTP/1.1
29…	32.257122440	81.31.186.20	192.168.43.152	HTTP	430	HTTP/1.1 302 Moved Temporarily (text/plai…
29…	32.264263759	192.168.43.152	81.31.186.20	HTTP	711	GET /c/portal/layout HTTP/1.1
29…	32.364922917	81.31.186.20	192.168.43.152	HTTP	417	HTTP/1.1 302 Moved Temporarily (text/html…
30…	32.372629834	192.168.43.152	81.31.186.20	HTTP	700	GET /home HTTP/1.1
30…	33.437378835	81.31.186.20	192.168.43.152	HTTP	713	HTTP/1.1 200 OK (text/html)
30…	33.468005187	192.168.43.152	81.31.186.20	HTTP	645	GET /html/portlet/ext/slide_content_image/…
30…	33.544278829	81.31.186.20	192.168.43.152	HTTP	355	HTTP/1.1 200 OK (text/html)
30…	33.638990295	192.168.43.152	81.31.186.20	HTTP	956	POST /c/portal/json_service HTTP/1.1 (app…
30…	33.711678462	81.31.186.20	192.168.43.152	HTTP	633	HTTP/1.1 200 OK (text/html)
31…	33.862592987	192.168.43.152	81.31.186.20	HTTP	956	POST /c/portal/json_service HTTP/1.1 (app…
31…	33.929629656	81.31.186.20	192.168.43.152	HTTP	624	HTTP/1.1 200 OK (text/html)
31…	34.017570304	192.168.43.152	81.31.186.20	HTTP	885	GET /html/js/editor/editor.jsp?p_l_id=1450…
31…	34.065649142	81.31.186.20	192.168.43.152	HTTP	13…	HTTP/1.1 200 OK (text/html)

شکل ۱: خروجی wireshark برای سایت old.sharif.edu

۱.۱ سوال اول

همانطور که در شکل ۲ مشاهده می‌شود بیشترین پروتوکل مورد استفاده TCP می‌باشد که ۹۷.۳ درصد از استفاده را به خود اختصاص داده است. با این حال مشاهده می‌شود که فقط ۰.۱ درصد از استفاده مربوط به داده‌های HTTP می‌باشد. علت این واقع باز بودن سایت‌ها و اپلیکیشن‌های دیگر در هنگام آزمایش بوده است.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End P
Frame	100.0	20414	100.0	6504666	162 k	0
Ethernet	100.0	20414	4.4	285796	7,135	0
Internet Protocol Version 4	99.8	20374	6.3	407480	10 k	0
User Datagram Protocol	2.5	503	0.1	4024	100	0
Simple Service Discovery Protocol	0.1	12	0.0	2076	51	12
Multicast Domain Name System	0.1	21	0.0	1617	40	21
Domain Name System	1.6	325	0.2	14448	360	325
Data	0.7	145	2.8	181250	4,525	145
Transmission Control Protocol	97.3	19871	86.2	5606855	139 k	1406
Transport Layer Security	28.6	5842	91.7	5963416	148 k	5786
Hypertext Transfer Protocol	0.1	16	0.3	22724	567	6
Line-based text data	0.0	8	1.2	76188	1,902	8
HTML Form URL Encoded	0.0	2	0.0	454	11	2
Address Resolution Protocol	0.2	40	0.0	1120	27	40

شکل ۲: آمارهای مشاهده شده در سوال ۱ آزمایش ۱ قبل از فیلتر کردن توسط http

در صورتی که ابتدا با HTTP بسته‌ها را فیلتر کنیم و پس از آن آمار را تماشا کنیم، متوجه می‌شویم که ۱۰۰ درصد بسته‌ها از نوع HTTP می‌باشند. برای دیدن نتایج می‌توانید به شکل ۳ مراجعه کنید.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packe
Frame	100.0	16	100.0	11133	38 k	0
Ethernet	100.0	16	2.0	224	765	0
Internet Protocol Version 4	100.0	16	2.9	320	1,094	0
Transmission Control Protocol	100.0	16	95.1	10589	36 k	0
Hypertext Transfer Protocol	100.0	16	204.1	22724	77 k	6
Line-based text data	50.0	8	684.3	76188	260 k	8
HTML Form URL Encoded	12.5	2	4.1	454	1,552	2

شکل ۳: آمارهای مشاهده شده در سوال ۱ آزمایش ۱ پس از فیلتر کردن توسط http

۲.۱ سوال دوم

مطابق شکل ۱ متوجه می‌شویم که زمان دریافت جواب حدودا برابر با ۰.۱۲ ثانیه می‌باشد. همچنین با توجه به شکل ۴ می‌توان متوجه شد که seq number برای اولین درخواست TCP برابر با ۳۶۱۴۹۵۵۴۳۹ بوده است. (برای دریافت این تنظیمات روی یک packet کلیک کرده و از زیرمنوی follow گزینه‌ی TCP Stream زا انتخاب کردیم)

tcp.stream eq 275

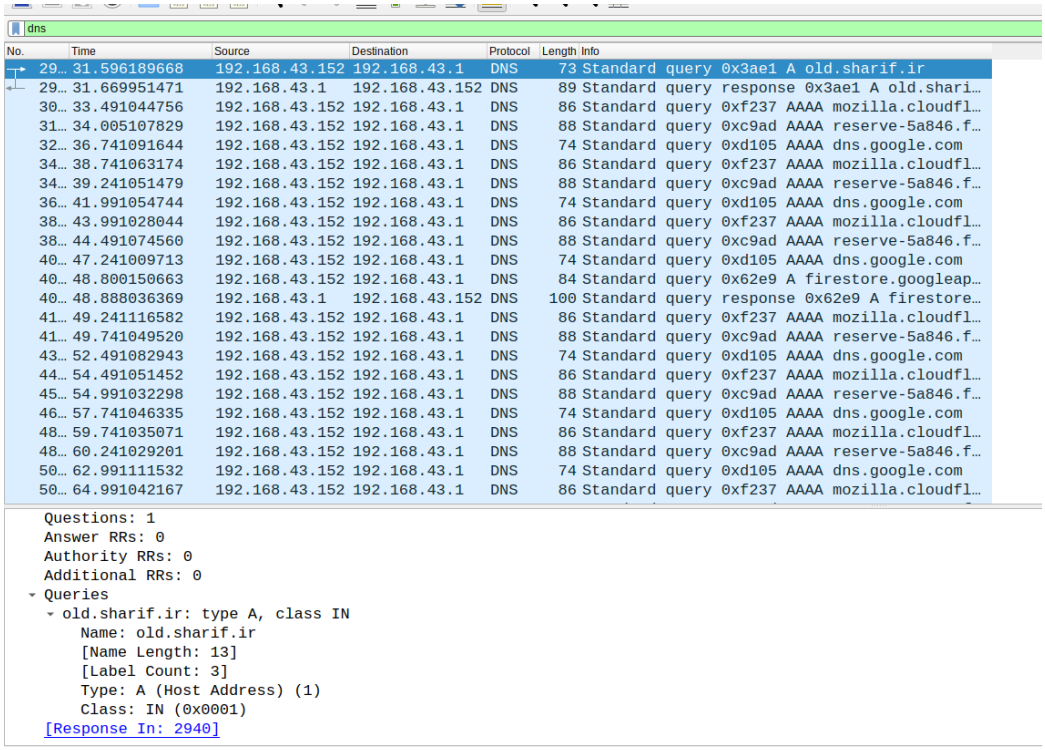
No.	Time	Source	Destination	Protocol	Length	Info
29…	31.670281715	192.168.43.152	81.31.186.20	TCP	74	36470 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS…
29…	31.725409709	81.31.186.20	192.168.43.152	TCP	74	80 → 36470 [SYN, ACK] Seq=0 Ack=1 Win=1448…
29…	31.725473765	192.168.43.152	81.31.186.20	TCP	66	36470 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len…
29…	31.725828263	192.168.43.152	81.31.186.20	HTTP	664	GET / HTTP/1.1
29…	31.770094111	81.31.186.20	192.168.43.152	TCP	66	80 → 36470 [ACK] Seq=1 Ack=599 Win=94208 L…
29…	31.847509031	81.31.186.20	192.168.43.152	HTTP	435	HTTP/1.1 200 OK (text/html)
29…	31.847556697	192.168.43.152	81.31.186.20	TCP	66	36470 → 80 [ACK] Seq=599 Ack=370 Win=64000…
29…	32.147975168	192.168.43.152	81.31.186.20	HTTP	697	GET /c HTTP/1.1
29…	32.257122440	81.31.186.20	192.168.43.152	HTTP	430	HTTP/1.1 302 Moved Temporarily (text/plai…
29…	32.257179683	192.168.43.152	81.31.186.20	TCP	66	36470 → 80 [ACK] Seq=1230 Ack=734 Win=6387…
29…	32.264263759	192.168.43.152	81.31.186.20	HTTP	711	GET /c/portal/layout HTTP/1.1
29…	32.364922917	81.31.186.20	192.168.43.152	HTTP	417	HTTP/1.1 302 Moved Temporarily (text/html…
30…	32.364967147	192.168.43.152	81.31.186.20	TCP	66	36470 → 80 [ACK] Seq=1875 Ack=1085 Win=638…
30…	32.372629834	192.168.43.152	81.31.186.20	HTTP	700	GET /home HTTP/1.1
30…	32.448120113	81.31.186.20	192.168.43.152	TCP	66	80 → 36470 [ACK] Seq=1085 Ack=2509 Win=977…
30…	33.423333382	81.31.186.20	192.168.43.152	TCP	28…	80 → 36470 [PSH, ACK] Seq=1085 Ack=2509 Wi…
30…	33.423333843	81.31.186.20	192.168.43.152	TCP	14…	80 → 36470 [ACK] Seq=3821 Ack=2509 Win=977…
30…	33.423391066	192.168.43.152	81.31.186.20	TCP	66	36470 → 80 [ACK] Seq=2509 Ack=3821 Win=614…
30…	33.423425107	192.168.43.152	81.31.186.20	TCP	66	36470 → 80 [ACK] Seq=2509 Ack=5189 Win=601…
30…	33.424462465	81.31.186.20	192.168.43.152	TCP	14…	80 → 36470 [PSH, ACK] Seq=5189 Ack=2509 Wi…
30…	33.424462726	81.31.186.20	192.168.43.152	TCP	28…	80 → 36470 [PSH, ACK] Seq=6557 Ack=2509 Wi…
30…	33.424462806	81.31.186.20	192.168.43.152	TCP	31…	80 → 36470 [PSH, ACK] Seq=9293 Ack=2509 Wi…
30…	33.424491698	192.168.43.152	81.31.186.20	TCP	66	36470 → 80 [ACK] Seq=2509 Ack=6557 Win=588…

```
▸ Frame 2941: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlp1s0, id 0
▸ Ethernet II, Src: 4c:79:6e:5b:43:48 (4c:79:6e:5b:43:48), Dst: 0a:c5:e1:90:6f:83 (0a:c5:e1:90:6f:83)
▸ Internet Protocol Version 4, Src: 192.168.43.152, Dst: 81.31.186.20
▾ Transmission Control Protocol, Src Port: 36470, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 36470
    Destination Port: 80
    [Stream index: 275]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 3614955439
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 0
```

شکل ۴: خروجی TCP Stream که به ما این قابلیت را می‌دهد تا به صورت ریز خروجی‌ها را مشاهده کنیم.

۳.۱ سوال سوم

طبق شکل‌های ۵ و ۶ مشاهده می‌شود که هم درخواست و هم جواب هر دو از نوع A می‌باشند.



شکل ۵: درخواست DNS

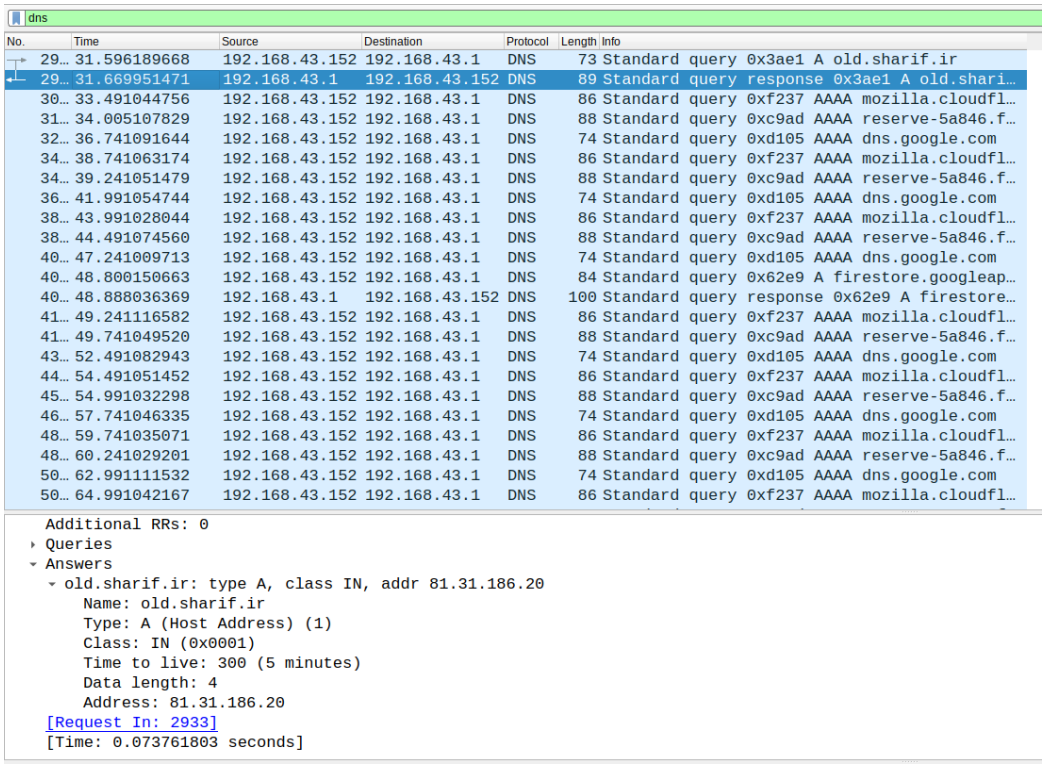
۴.۱ سوال چهارم

در این قسمت متوجه شدیم که در خروجی‌های قسمت قبل عکسی وجود ندارد. علت این واقعه به این دلیل است که احتمالا مرورگر عکس‌ها را در دفعات قبلی باز کردن سایت cache کرده بود. بنابراین با استفاده از کلید Ctrl+F۵ توانستیم cache را پاک کرده و عکس‌ها را از اول دریافت کنیم. سپس با استفاده از فیلتر image-jiff عکس‌های دریافت شده را فیلتر کردیم. عکس‌های دریافت شده در شکل ۷ دیده می‌شوند. با انتخاب یک عکس و انتخاب گزینه‌ی JPEG File Interchange Format مانند شکل ۸ و زدن دکمه‌ی Ctrl+X می‌توان خروجی عکس را دریافت کرد. همچنین خروجی در شکل ۹ دیده می‌شود.

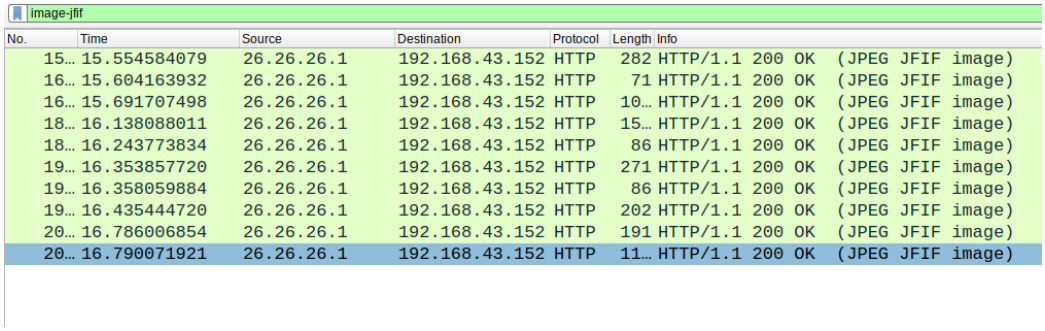
۲ بخش دوم

با استفاده از دستور telnet telehack.com در ترمینال ubuntu صفحه‌ای باز می‌شود که تعدادی دستور در ابتدا نمایش می‌دهد. این مشاهده‌ی اولیه را می‌توان در شکل ۱۰ تماشا کرد. پس از آن دستورهای ? و joke و qr امتحان شده که خروجی‌های آن‌ها به ترتیب در شکل ۱۱ و شکل ۱۲ و شکل ۱۳ مشاهده می‌شود.

در نرم‌افزار wireshark داده‌های telnet را فیلتر می‌کنیم. این داده‌ها در شکل ۱۴ مشاهده می‌شود. با بررسی داده‌های ارسال و دریافت شده متوجه می‌شویم که با ارسال یک داده همان داده را دریافت کرده‌ایم این نشان دهنده‌ی این است که پروتکل telnet در هنگام دریافت داده از سرور آن را به ما نمایش می‌دهد نه زمانی که داده‌ها توسط کاربر وارد می‌شود. در شکل ۱۵ داده‌ی ارسال شده دیده می‌شود و در شکل ۱۶ داده‌ی دریافت شده. همچنین در شکل ۱۷ مشاهده می‌شود پس از دریافت دستور وارد شده خروجی آن نیز ارسال شده است.



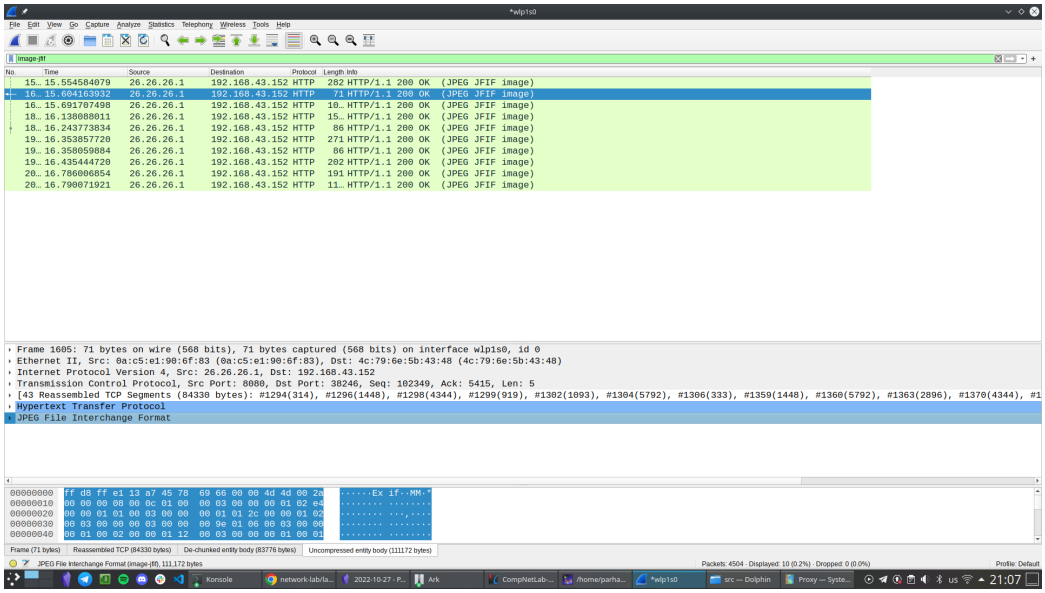
شکل ۶: جواب DNS



شکل ۷: عکس‌های دریافت شده

۱.۲ سوال اول

با باز کردن داده‌های گفته شده با تصویر ۱۸ مواجه می‌شویم. با توجه داده‌ها متوجه می‌شویم که آدرس مبدا برابر با ۱۹۲.۱۶۸.۰.۲ و آدرس مقصد برابر با ۱۹۲.۱۶۸.۰.۱ می‌باشد.



شکل ۸: مراحل خروجی گرفتن عکس

شکل ۹: عکس خروجی

۲.۲ سوال دوم

با راست کلیک روی بسته‌ی اول، زدن follow و TCP Stream می‌توانیم روند طی شده را مشاهده کنیم. خروجی در شکل ۱۹ مشاهده می‌شود. می‌توان متوجه شد که یوزرنیم کاربر برابر با fake بوده و رمز عبور وی برابر با user بوده است.

۳.۲ سوال سوم

دوباره با مراجعه به شکل ۱۹، می‌توان متوجه شد که دستورهای زده شده به ترتیب به صورت زیر می‌باشند: (بین دستور اول و دوم هم interrupt فرستاده است تا فرایند ping قطع شود)

```
parham@parham-laptop:~$ telnet telehack.com
Trying 64.13.139.230...
Connected to telehack.com.
Escape character is '^]'.

Connected to TELEHACK port 41

It is 10:59 am on Thursday, October 27, 2022 in Mountain View, California, USA.
There are 89 local users. There are 26642 hosts on the network.

  Type HELP for a detailed command list.
  Type NEWUSER to create an account.
  Press control-C to interrupt any command.

May the command line live forever.

Command, one of the following:
  2048         ?            ac           advent       aquarium     basic
  bf           c8           cal          calc         callsign     ching
  clear        clock        cowsay       date         ddate        echo
  eliza        factor       figlet       fnord        geoip        ipaddr
  joke         login        mac          md5          minesweeper  morse
  newuser      notes        octopus      phoon        pig          pong
  primes       privacy      qr           rain         rand         rfc
  rig          roll         rot13        salvo        sleep        starwars
  traceroute   typespeed    units        uptime       usenet       users
  uupath       uuplot       weather      when         zc           zork

.
```

شکل ۱۰: ورودی telnet

```
.?
Command, one of the following:
  2048         ?            a2           ac           advent       aquarium
  basic        bf           c8           cal          calc         callsign
  ching        clear        clock        cowsay       date         ddate
  echo         eliza        factor       figlet       finger       fnord
  geoip        gif          help         ipaddr       joke         login
  mac          md5          minesweeper  morse        newuser      notes
  octopus      phoon        pig          ping         pong         primes
  privacy      qr           rain         rand         rfc          rig
  roll         rot13        salvo        sleep        starwars     sudoku
  traceroute   typespeed    units        uptime       usenet       users
  uumap        uupath       uuplot       weather      when         zc
  zork
```

شکل ۱۱: خروجی دستور ?

/sbin/ping www.yahoo.com, ls, ls -a, exit

```
.joke
"You can call Usenet a democracy if you want to.  You can call it a
 totalitarian dictatorship run by space aliens and the ghost of Elvis.
 It doesn't matter either way."
```

شکل ۱۲: خروجی دستور joke

۳ بخش سوم

ما سایت github.com را انتخاب کردیم. در شکل ۲۰ می‌توان خروجی‌های مربوط به این دستور را در wireshark مشاهده کرد. شکل ۲۱ شامل هایflag دستور dns برای پکت ارسالی می‌باشد و شکل ۲۲ شامل هایflag دستور dns برای پکت دریافتی می‌باشد. همچنین در شکل ۲۳ می‌توانید نتیجه‌ی اجرای دستور dig در ترمینال را مشاهده کنید.

۱.۳ سوال اول

همانطور که در شکل ۲۰ مشاهده می‌شود، این درخواست برای سرور ۱۹۲.۱۶۸.۴۳.۱ ارسال می‌شود و جواب نیز از آن دریافت می‌شود. علت این است که در تنظیمات مربوط به کامپیوتر این سرور لوکال برای dns ذخیره شده است.

۲.۳ سوال دوم

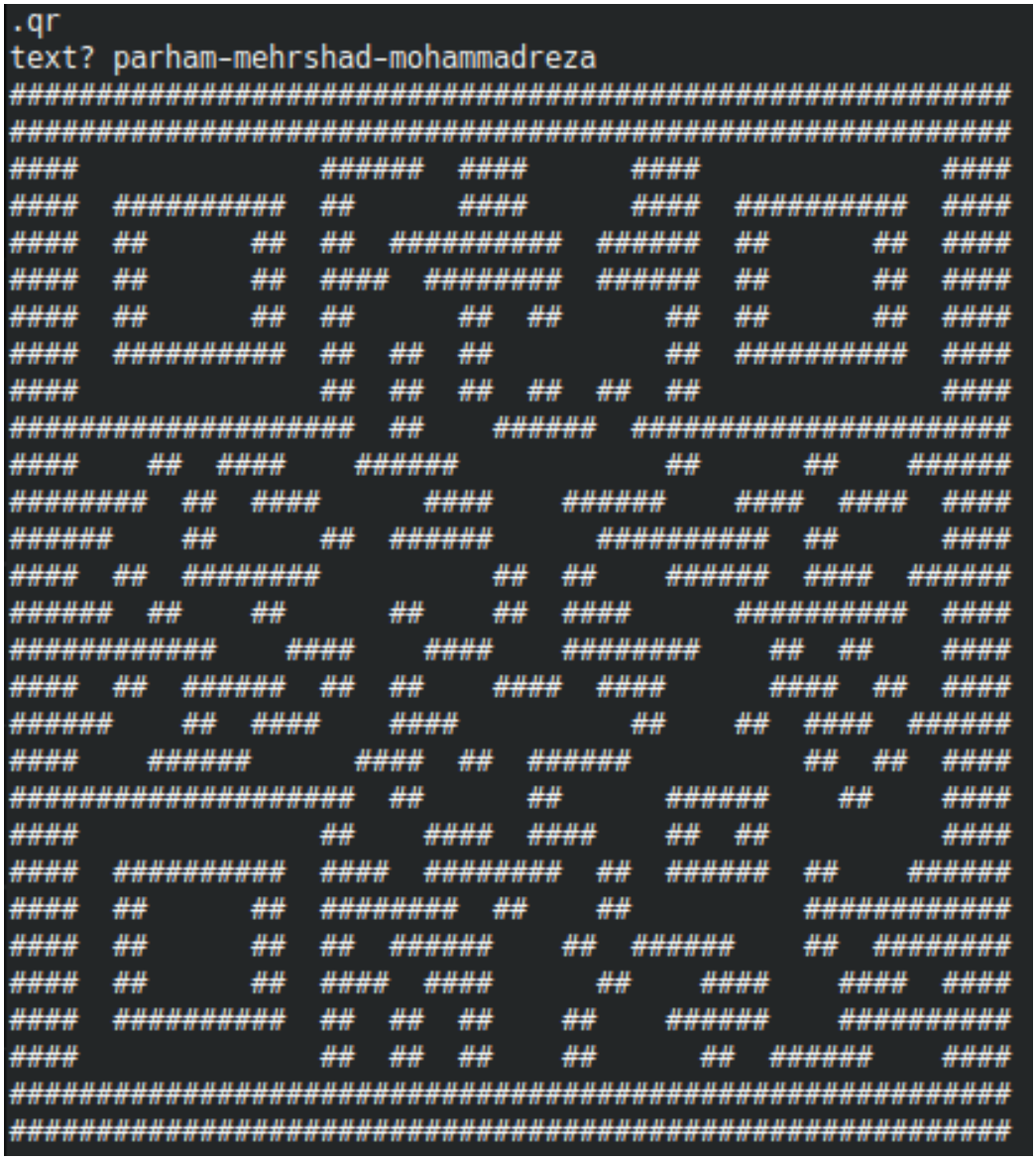
این سوال را با توجه به شکل‌های ۲۱ و ۲۲ پاسخ می‌دهیم. ابتدا لازم به ذکر است که هدرهای یک پیام dns شامل بخش‌های مختلفی است این بخش‌ها عبارتند از: Trunction ID، Flags، Questions، Answer RRs، Authority RRs، Additional RRs، Queries که حال به بررسی آن‌ها می‌پردازیم. مورد اول که TID می‌باشد شماره‌ی مخصوصی است که مشخص می‌کند یک جواب مربوط به چه سوالی است. پرچم‌ها را در انتها بررسی می‌کنیم. قسمت بعدی سوال‌هاست که اینجا عدد آن برابر یک است. قسمت بعدی جواب‌هاست که برای کوئری این عدد برابر ۰ است و برای پاسخ این عدد برابر با ۱ است.

قسمت Authority و Additional برای هر دو یکسان است. در قسمت Queries که این هم یکسان است درخواست ارسالی به سرور را مشاهده می‌کنیم که گیتهاب است. برای پاسخ قسمت Answers نیز داریم که جواب دریافت شده از سرور را نشان می‌دهد، اطلاعات مهمی که در آن حاضر است عبارت است از زمان اعتبار داشتن جواب، آی‌پی، و نام می‌باشد.

پرچم‌ها نیز قسمت‌های یکسان زیادی دارند ابتدا آن‌ها را نام می‌بریم و توضیح می‌دهیم. اولین پرچم مشخص می‌کند که این پکت query است یا answer. در ادامه truncated نشان می‌دهد که بسته کامل است یا نه که در این مثال هر دو بسته کامل می‌باشند. recursion مشخص می‌کند که آیا سرور باید درخواست را به سرورهای دیگر بفرستد یا نه که برای پاسخ و جواب recursion desired داریم که نشان می‌دهد آیا سرور مجاز به این کار است یا نه. در پاسخ یک پرچم اضافی در این مورد داریم به نام recursion available که نشان می‌دهد آیا سرور توانایی این کار را دارد یا نه.

پرچم دیگری که در هر دو یکسان است non-authenticated-date است که نشان می‌دهد آیا پاسخ authenticate نشده قابل قبول است یا نه که در این مثال برای هر دو غیر قابل قبول مشخص شده.

در نهایت replay-code مشخص می‌کند که وضعیت پاسخ به چه گونه است. در این مثال خطایی وجود نداشته و وضعیت با ۰ گزارش شده است.



شکل ۱۳: خروجی دستور telqr با استفاده از متن parham-mehrshad-mohammadreza

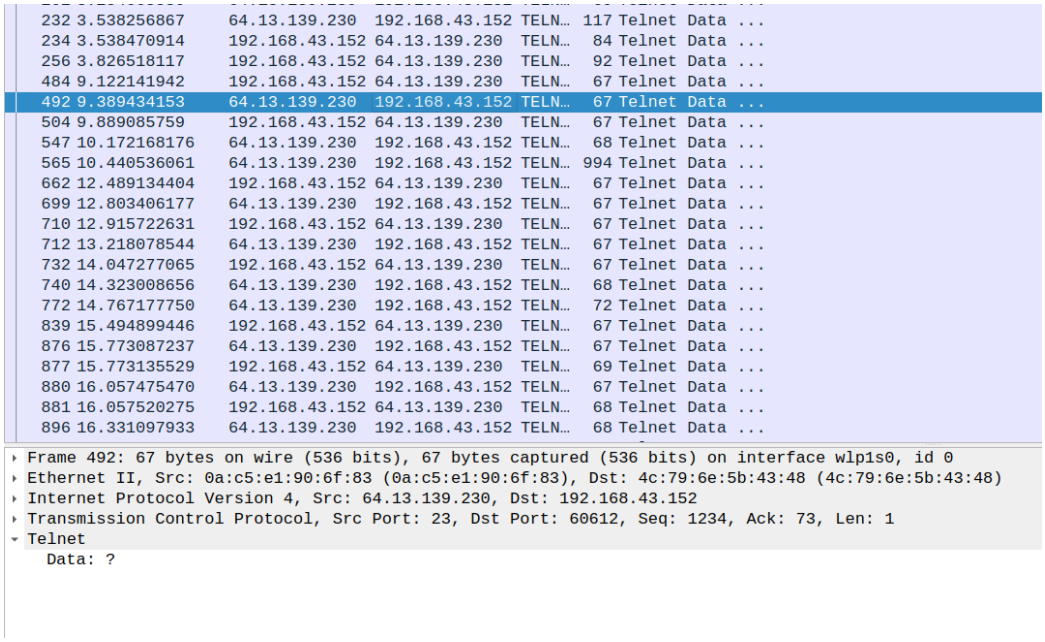
173	2.983112595	192.168.43.152	64.13.139.230	TELN…	93	Telnet	Data ...
202	3.254665530	64.13.139.230	192.168.43.152	TELN…	69	Telnet	Data ...
232	3.538256867	64.13.139.230	192.168.43.152	TELN…	117	Telnet	Data ...
234	3.538470914	192.168.43.152	64.13.139.230	TELN…	84	Telnet	Data ...
256	3.826518117	192.168.43.152	64.13.139.230	TELN…	92	Telnet	Data ...
484	9.122141942	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...
492	9.389434153	64.13.139.230	192.168.43.152	TELN…	67	Telnet	Data ...
504	9.889085759	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...
547	10.172168176	64.13.139.230	192.168.43.152	TELN…	68	Telnet	Data ...
565	10.440536061	64.13.139.230	192.168.43.152	TELN…	994	Telnet	Data ...
662	12.489134404	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...

شکل ۱۴: داده‌های telnet در wireshark

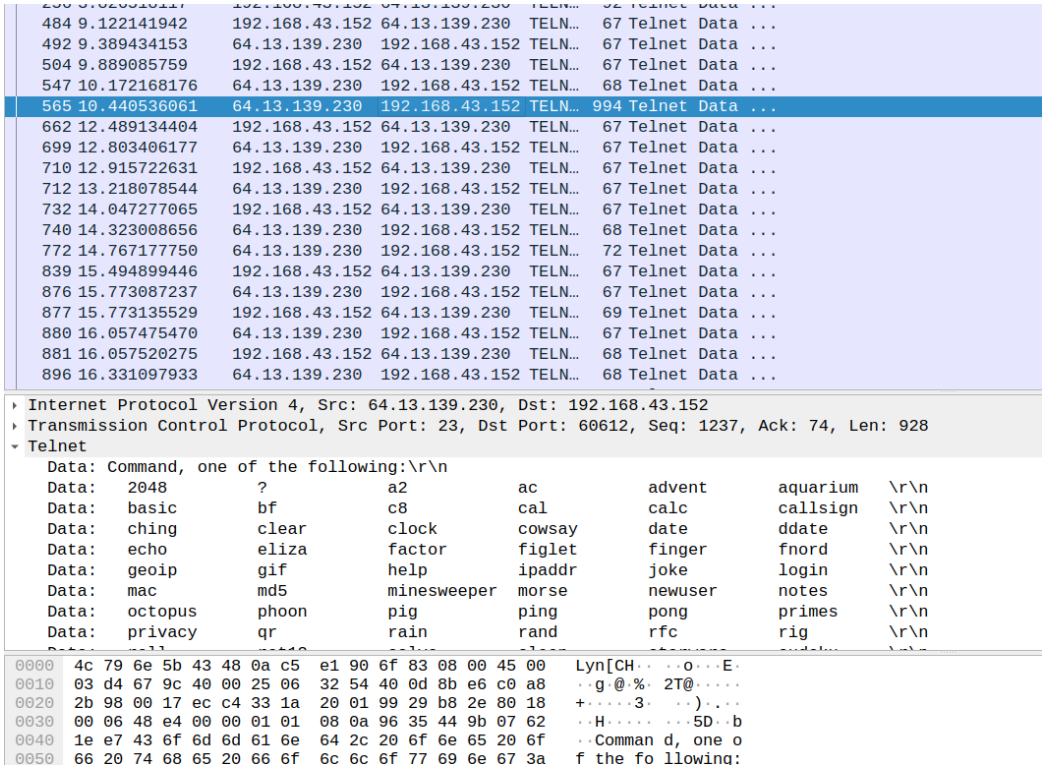
232	3.538256867	64.13.139.230	192.168.43.152	TELN…	117	Telnet	Data ...
234	3.538470914	192.168.43.152	64.13.139.230	TELN…	84	Telnet	Data ...
256	3.826518117	192.168.43.152	64.13.139.230	TELN…	92	Telnet	Data ...
484	9.122141942	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...
492	9.389434153	64.13.139.230	192.168.43.152	TELN…	67	Telnet	Data ...
504	9.889085759	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...
547	10.172168176	64.13.139.230	192.168.43.152	TELN…	68	Telnet	Data ...
565	10.440536061	64.13.139.230	192.168.43.152	TELN…	994	Telnet	Data ...
662	12.489134404	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...
699	12.803406177	64.13.139.230	192.168.43.152	TELN…	67	Telnet	Data ...
710	12.915722631	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...
712	13.218078544	64.13.139.230	192.168.43.152	TELN…	67	Telnet	Data ...
732	14.047277065	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...
740	14.323008656	64.13.139.230	192.168.43.152	TELN…	68	Telnet	Data ...
772	14.767177750	64.13.139.230	192.168.43.152	TELN…	72	Telnet	Data ...
839	15.494899446	192.168.43.152	64.13.139.230	TELN…	67	Telnet	Data ...
876	15.773087237	64.13.139.230	192.168.43.152	TELN…	67	Telnet	Data ...
877	15.773135529	192.168.43.152	64.13.139.230	TELN…	69	Telnet	Data ...
880	16.057475470	64.13.139.230	192.168.43.152	TELN…	67	Telnet	Data ...
881	16.057520275	192.168.43.152	64.13.139.230	TELN…	68	Telnet	Data ...
896	16.331097933	64.13.139.230	192.168.43.152	TELN…	68	Telnet	Data ...

```
▸ Frame 484: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface wlp1s0, id 0
▸ Ethernet II, Src: 4c:79:6e:5b:43:48 (4c:79:6e:5b:43:48), Dst: 0a:c5:e1:90:6f:83 (0a:c5:e1:90:6f:83)
▸ Internet Protocol Version 4, Src: 192.168.43.152, Dst: 64.13.139.230
▸ Transmission Control Protocol, Src Port: 60612, Dst Port: 23, Seq: 72, Ack: 1234, Len: 1
▾ Telnet
    Data: ?
```

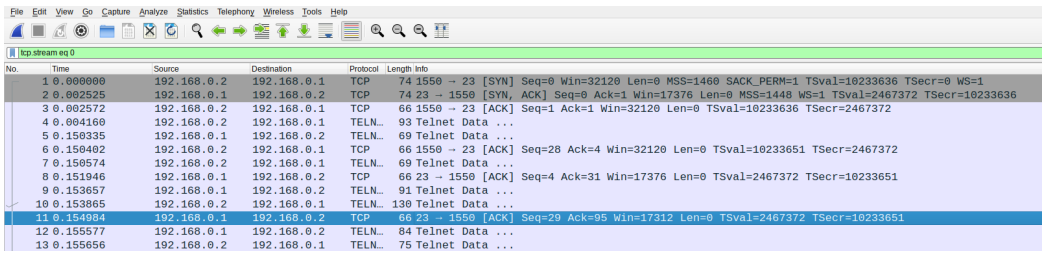
شکل ۱۵: کاراکتر ارسالی



شکل ۱۶: کاراکتر دریافتی

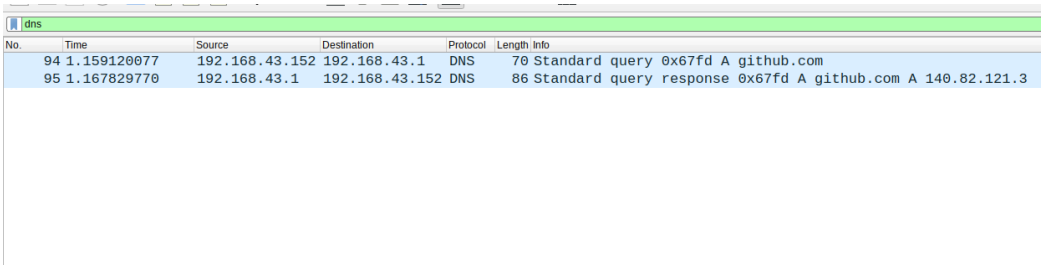


شکل ۱۷: جواب دریافتی

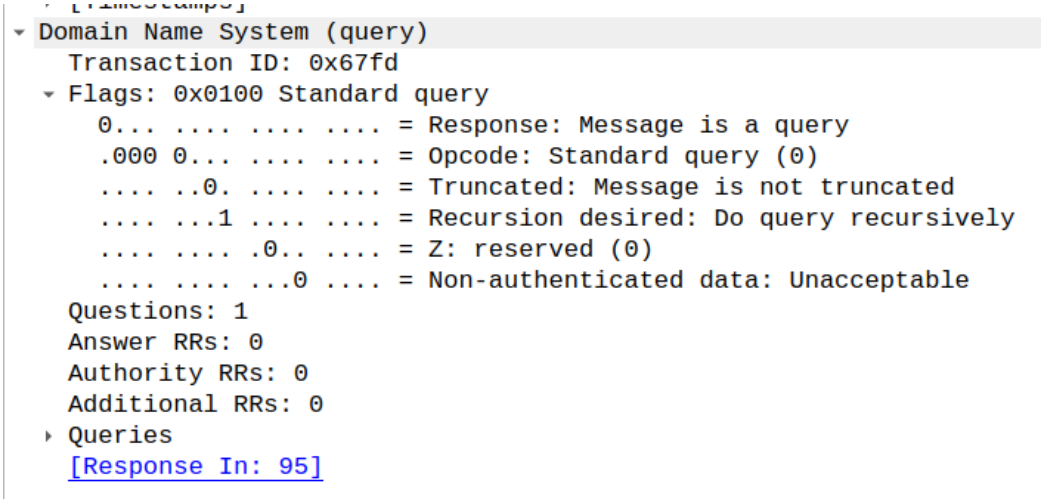


شکل ۱۸: داده‌های مشاهده شده در فایل telnet.pcap

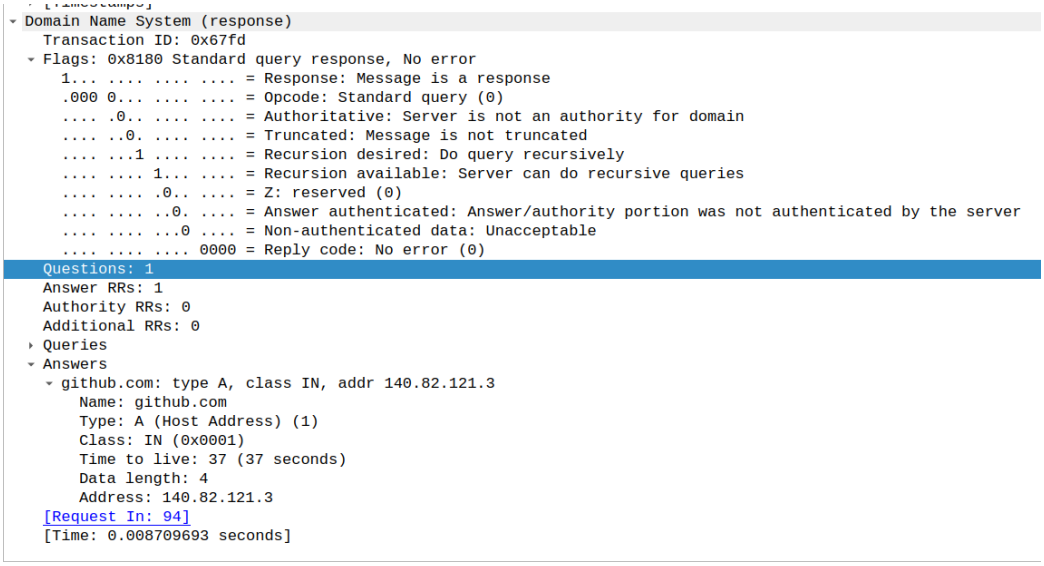
شکل ۱۹: داده‌های وارد شده و دریافت شده توسط کاربر



شکل ۲۰: خروجی‌های دستور dig در نرم‌افزار wireshark



شکل ۲۱: پرچم‌های سوال dns در wireshark



شکل ۲۲: پرچم‌های dns mhso در wireshark

```
parham@parham-laptop:~$ dig github.com

; <<>> DiG 9.16.1-Ubuntu <<>> github.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16531
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;github.com.                    IN      A

;; ANSWER SECTION:
github.com.             37      IN      A       140.82.121.3

;; Query time: 7 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: 2022 0330+ 01:00:37 28 جؤمعه اؤکتؤبر
;; MSG SIZE  rcvd: 55
```

شکل ۲۳: خروجی دستور dig در ترمینال