



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

آزمایشگاه شبکه

سپهر مقیسه

شماره دانشجویی: ۹۸۳۱۱۰۳

زمستان ۱۴۰۰

۱. همانطور که مشاهده میشود از پروتوکول udp استفاده شده است:

27171 120.638015	192.168.1.51	190.2.133.223	UDP	200 60579 → 80 Len=158
27172 121.190571	192.168.1.51	190.2.133.223	UDP	108 60579 → 80 Len=66
27173 121.193038	190.2.133.223	192.168.1.51	UDP	107 80 → 60579 Len=65
27174 121.193038	190.2.133.223	192.168.1.51	UDP	158 80 → 60579 Len=116
27175 121.206033	192.168.1.51	190.2.133.223	UDP	560 60579 → 80 Len=518
27176 121.645248	190.2.133.223	192.168.1.51	UDP	119 80 → 60579 Len=77
27177 121.645248	190.2.133.223	192.168.1.51	UDP	107 80 → 60579 Len=65
27178 121.645248	190.2.133.223	192.168.1.51	UDP	1433 80 → 60579 Len=1391
27179 121.645529	190.2.133.223	192.168.1.51	UDP	347 80 → 60579 Len=305
27180 121.646860	192.168.1.51	190.2.133.223	UDP	107 60579 → 80 Len=65
27181 122.056823	192.168.1.51	190.2.133.223	UDP	108 60579 → 80 Len=66
27182 122.567111	190.2.133.223	192.168.1.51	UDP	119 80 → 60579 Len=77

Frame 27175: 560 bytes on wire (4480 bits), 560 bytes captured (4480 bits) on interface \Device\NPF_{62E330ED-9DB4-48F7-AFC6-DB5A08} Ethernet II, Src: AzureWav_9a:eb:5d (dc:f5:05:9a:eb:5d), Dst: KZBroadb_fe:0a:f3 (6c:ad:ef:fe:0a:f3)

Internet Protocol Version 4, Src: 192.168.1.51, Dst: 190.2.133.223

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 546

Identification: 0x263e (9790)

> Flags: 0x00

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 128

Protocol: UDP (17)

Header Checksum: 0x0cd0 [validation disabled]

۲. در لایه اول از Ethernet، لایه دوم ipv4 و در لایه سوم udp:

> Frame 27175: 560 bytes on wire (4480 bits), 560 bytes captured (4480 bits) on interface \Device\NPF_{62E330ED-9DB4-48F7-AFC6-DB5A08}

> Ethernet II, Src: AzureWav_9a:eb:5d (dc:f5:05:9a:eb:5d), Dst: KZBroadb_fe:0a:f3 (6c:ad:ef:fe:0a:f3)

> Destination: KZBroadb_fe:0a:f3 (6c:ad:ef:fe:0a:f3)

> Source: AzureWav_9a:eb:5d (dc:f5:05:9a:eb:5d)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 192.168.1.51, Dst: 190.2.133.223

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 546

Identification: 0x263e (9790)

> Flags: 0x00

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 128

Protocol: UDP (17)

Header Checksum: 0x0cd0 [validation disabled]

[Header checksum status: Unverified]

همانطور که میدانیم، در سیستم لایه ای، هر لایه در ابتدای بسته ی لایه بالایی خود اطلاعاتی را به عنوان هدر اضافه میکند و سپس به لایه ی زیرین منتقل می کند تا در نهایت توسط لایه ی فیزیکی منتقل شود. در نتیجه بیت ها در شروع هر پکت، ابتدا متعلق به هدرهای لایه ۱ و پس از آن به ترتیب مربوط به لایه های ۳ و ۴ هستند.

در پکت مشاهده شده می بینیم که ۵۶۰ بایت در مجموع اندازه پکت بوده است. در لایه ی دوم نیز در بخش total length مشاهده می کنیم که اندازه پکت لایه ی دوم ۵۴۶ بایت است

۳. بله. بسته های پروتکل ARP که برای پیدا کردن مک آدرس دیگر دستگاه های داخلی هستند، تنها از ۲ لایه ابتدایی استفاده می کنند.

```
> Frame 18882: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{62E330ED-9DB4-48F7-AFC6-DB5A080E5D40}, id 0
v Ethernet II, Src: AzureWav_9a:eb:5d (dc:f5:05:9a:eb:5d), Dst: KZBroadb_fe:0a:f3 (6c:ad:ef:fe:0a:f3)
  v Destination: KZBroadb_fe:0a:f3 (6c:ad:ef:fe:0a:f3)
    Address: KZBroadb_fe:0a:f3 (6c:ad:ef:fe:0a:f3)
      ....0. .... = LG bit: Globally unique address (factory default)
      ....0. .... = IG bit: Individual address (unicast)
  > Source: AzureWav_9a:eb:5d (dc:f5:05:9a:eb:5d)
    Type: ARP (0x0806)
> Address Resolution Protocol (request)
```

```
Identification: 0x2dec (11756)
Flags: 0x00
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: UDP (17)
Header Checksum: 0x06e7 [validation disabled
[Header checksum status: Unverified]
Source Address: 192.168.1.51
Destination Address: 190.2.133.223
er Datagram Protocol, Src Port: 60579, Dst Po
۴.
۵. پورت مبدا و مقصد در شکل زیر مشخص شده است
```

```
> User Datagram Protocol, Src Port: 60579, Dst Port: 80
```

در خدماتی که IP ارائه میدهد، برای تمیز دادن سرویس های مختلف بر روی یک آدرس IP، از شماره پورت در کنار آدرس IP استفاده میشود. در واقع هر انتقال اطلاعات در بستر IP، با زوج port:ip مشخص میشود و اطلاعات از یک پورت مبدا با آدرس IP مشخص به پورتی واقع در مقصد که آدرس IP خود را دارد منتقل می شود

```
Checksum: 0x895d [unverified]
```

۶. از udp استفاده شده است

```
> Internet Protocol Version 4, Src: 192.168.1.51, Dst: 178.22.122.100
```

۷. آدرس 192.168.1.51

Queries

> functional.events.data.microsoft.com: type A, class IN

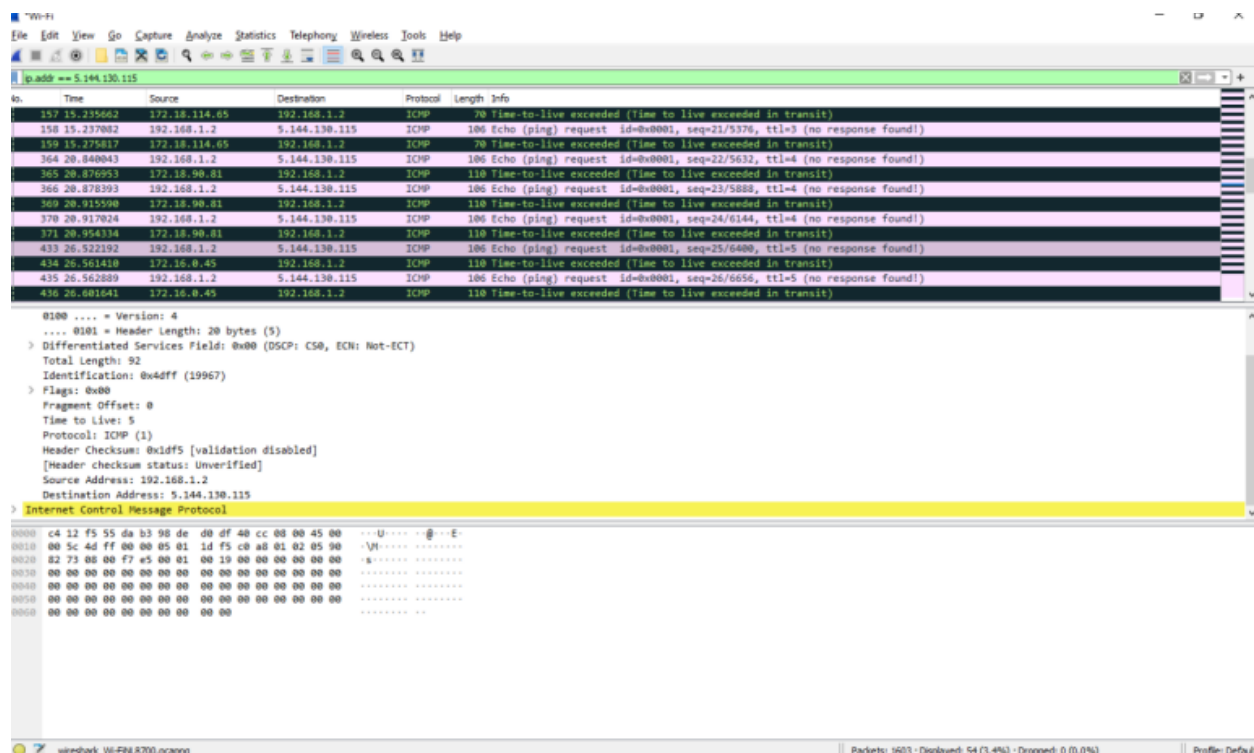
[Response In: 76]

نوع رکورد کوئری آن A است. این رکورد برای تبدیل نام هوست به آدرس ۳۲ بیتی IPv4 هوست استفاده می شود

۹. در این کوئری تایپ PTR استفاده شده است. این رکورد برای تبدیل آدرس IP به نام دامنه در reverse DNS lookup استفاده می شود. در اینجا نیز برای پیدا کردن نام دامنه‌ای که آدرس IP برابر با ۱,۱,۱,۱ دارد توسط دستور nslookup استفاده شده است

۱۰. پروتکل DNS رکوردهای متنوعی دارد که برخی از آنها عبارتند از MX، TXT و AAAA

۱۱. با اعمال این فیلتر، تنها بسته هایی که آدرس IP مبدا یا مقصد آنها برابر 5.144.130.115 است نمایش داده می شوند. این بسته ها که توسط tracer ارسال می شوند، همگی از پروتکل ICMP استفاده میکنند



Wireshark packet capture showing ICMP Echo (ping) requests. The packet list shows 11 requests, all with 'Time-to-live exceeded' status. The packet details show the ICMP Echo request structure with TTL=5. The packet bytes show the raw ICMP data.

۱۲. مقدار تایپ اولین بسته ۸ (echo (ping) request) و مقدار TTL آن برابر ۱ است.

۱۳. مقدار TTL بسته ها یک واحد یک واحد در حال افزایش است. در واقع این ابزار ابتدا ۳ بسته ICMP با $TTL=1$ ارسال می کند تا در اولین hop بازگشت داده شود. سپس ۳ بسته با $TTL=2$ ارسال می کند تا در دومین hop برگردانده شود. به همین ترتیب مقدار TTL را افزایش می دهد تا زمانی که دیگر خطایی در خصوص TTL گزارش نشود. با این کار hop های موجود در مسیر تا رسیدن به مقصد مشخص می شوند.

۱۴. پروتکل های IP دارای شمارهای هستند که در [این لینک](#) لیست آنها را مشاهده میکنیم. همانطور که در این لیست میبینیم، پروتکل TCP دارای شماره ۶ است. در نتیجه با اعمال فیلتر $ip.proto==6$ ، تنها بستههای TCP را در Wireshark خواهیم دید