Wireshark: Inspecting UDP Packets

Rwithik Manoj, Roll No: 53

Aim

Using Wireshark observe data transferred in client server communication using UDP and identify the UDP datagram.

Theory

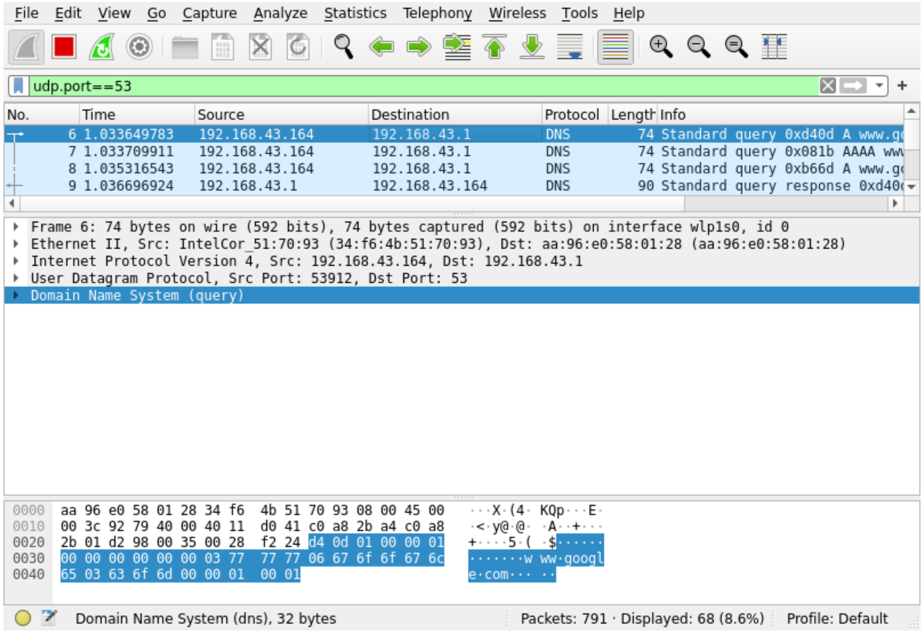
Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possi- ble. Some of the main uses include:

- Network administrators use it to troubleshoot network problems
- Developers use it to debug protocol implementations
- QA engineers use it to verify network applications
- Network security engineers use it to examine security problems
- People use it to learn network protocol internals

Output

In the above image, the highlighted part in the hexdump represents the UDP datagram:

- Source port: `d2 98` *rarrow* `53912`
- Destination port: `00 35` *rarrow* `53`
- UDP Length: `00 28` *rarrow* `40`
- UDP Checksum: `f2 24`



```
udp.port==53
```

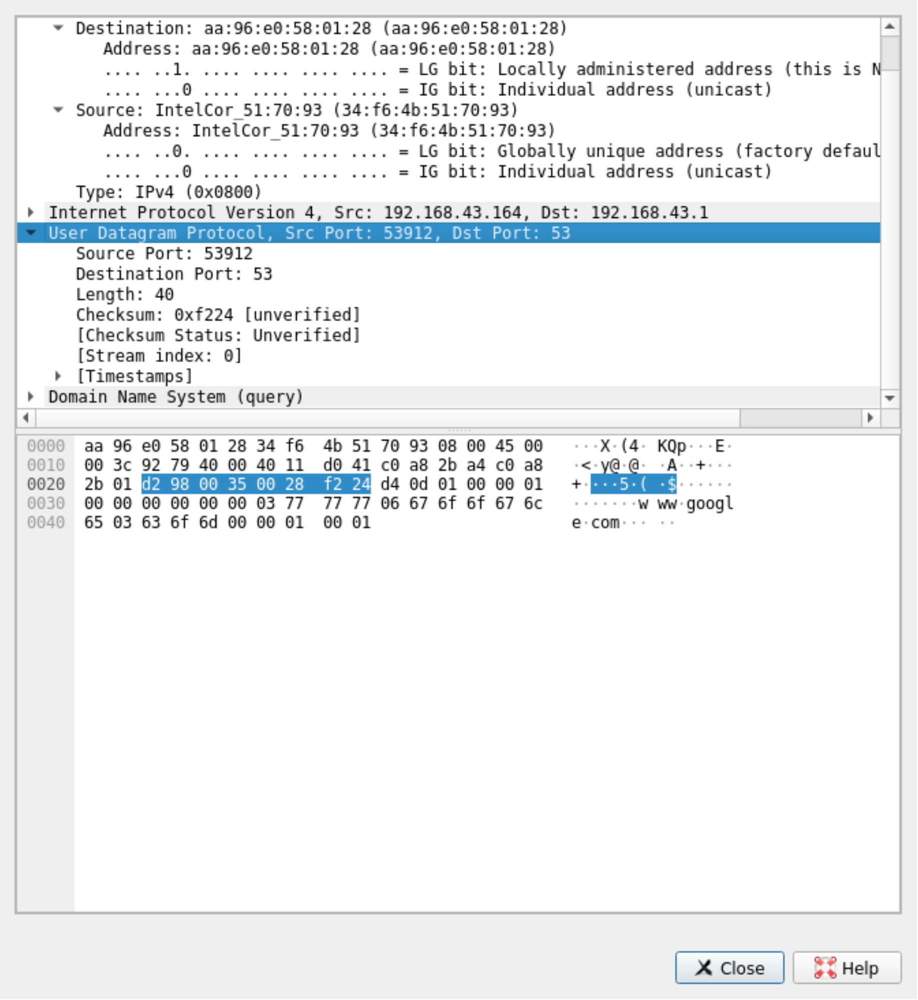
```
No.   Time          Source           Destination      Protocol  Length Info
    6 1.033649783   192.168.43.164   192.168.43.1     DNS          74 Standard query 0xd40d A www.gc
    7 1.033709911   192.168.43.164   192.168.43.1     DNS          74 Standard query 0x081b AAAA www
    8 1.035316543   192.168.43.164   192.168.43.1     DNS          74 Standard query 0xb66d A www.gc
    9 1.036696924   192.168.43.1     192.168.43.164   DNS          90 Standard query response 0xd40c
```

```
Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlp1s0, id 0
Ethernet II, Src: IntelCor_51:70:93 (34:f6:4b:51:70:93), Dst: aa:96:e0:58:01:28 (aa:96:e0:58:01:28)
Internet Protocol Version 4, Src: 192.168.43.164, Dst: 192.168.43.1
User Datagram Protocol, Src Port: 53912, Dst Port: 53
Domain Name System (query)
```

```
0000  aa 96 e0 58 01 28 34 f6  4b 51 70 93 08 00 45 00   ···X·(4· KQp···E·
0010  00 3c 92 79 40 00 40 11  d0 41 c0 a8 2b a4 c0 a8   ·<·y@·@· ·A··+···
0020  2b 01 d2 98 00 35 00 28  f2 24 d4 0d 01 00 00 01   +····5·( ·$······
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   ·······w ww·googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e·com··· ··
```

Domain Name System (dns), 32 bytes   Packets: 791 · Displayed: 68 (8.6%)   Profile: Default

Figure 1: UDP packet used for DNS query



```
    Destination: aa:96:e0:58:01:28 (aa:96:e0:58:01:28)
        Address: aa:96:e0:58:01:28 (aa:96:e0:58:01:28)
        .... ..1. .... .... .... .... = LG bit: Locally administered address (this is N
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: IntelCor_51:70:93 (34:f6:4b:51:70:93)
        Address: IntelCor_51:70:93 (34:f6:4b:51:70:93)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory defaul
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.43.164, Dst: 192.168.43.1
User Datagram Protocol, Src Port: 53912, Dst Port: 53
    Source Port: 53912
    Destination Port: 53
    Length: 40
    Checksum: 0xf224 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Timestamps]
Domain Name System (query)
```

```
0000  aa 96 e0 58 01 28 34 f6  4b 51 70 93 08 00 45 00   ···X·(4· KQp···E·
0010  00 3c 92 79 40 00 40 11  d0 41 c0 a8 2b a4 c0 a8   ·<·y@·@· ·A··+···
0020  2b 01 d2 98 00 35 00 28  f2 24 d4 0d 01 00 00 01   +····5·( ·$······
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   ·······w ww·googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e·com··· ··
```

Figure 2: Details of the Packet