

DNS Traffic:

DNS stand for Domain Name System, it's a protocol that resolve name to IP addresses.

Few things about DNS:

- DNS is a query-response protocol
- DNS traffic normally use UDP port 53
- DNS traffic should go to DNS servers only

Some facts that can help us distinguish Normal and Suspicious DNS Traffic:

Normal DNS Traffic	Suspicious DNS Traffic
Port 53, UDP	Traffic on port 53 but using TCP instead of UDP
Should only go to DNS Servers	DNS traffic not going to DNS Servers
Should see DNS Responses to DNS Queries	A lot of DNS Queries with no DNS responses or vice versa

So if we see DNS other than 53 port which is normally the UDP protocol, but if its 53 tcp its either DNSSEC or its malicious. If we see the DNS traffic is going to an IP that is not a DNS server then its more likely a C2 connection. And if we see there is a lot of DNS queries made to outside but no response is sent back to those queries(maybe DATA exfiltration where the receiving point just receive the data through the Queries made to that server and those queries will have the data in it.)

Normal DNS:

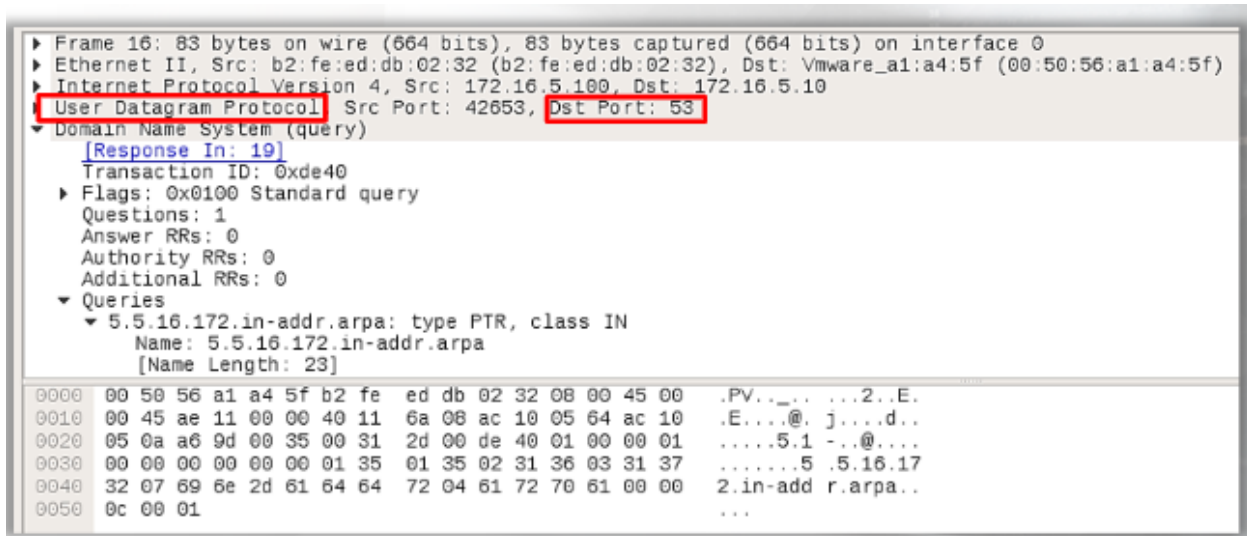
dns						
No.	Time	Source	Destination	Protocol	Length	Info
16 26	200151138	172.16.5.100	172.16.5.10	DNS	83	Standard query 0xde40 PTR 5.5.16.172.in-addr.arpa
19 26	272980431	172.16.5.10	172.16.5.100	DNS	127	Standard query response 0xde40 PTR 5.5.16.172.in-addr.arpa PTR wkst-techsupport.sportsfoo.com
41 56	605405613	172.16.5.100	172.16.5.10	DNS	94	Standard query 0xa620 PTR 5.5.16.172.in-addr.arpa OPT
42 56	639661726	172.16.5.10	172.16.5.100	DNS	138	Standard query response 0xa620 PTR 5.5.16.172.in-addr.arpa PTR wkst-techsupport.sportsfoo.com OPT

Here we see 4 packets, 2 are DNS Queries and 2 are DNS responses, and notice that there is 2 deferent Transaction IDs in both Query and in the Response the 0xde40 and 0xa620, so that mean the Request Query and the Response to that Query should have the same Transaction ID. Mean we make a DNS Query that has a Transaction ID, and when we get a response to this Query, it will have that Transaction ID in it and that's how we know to which Query Request this response is for.

DNS transaction ID:

Its 16-bit that identify specific Transaction mean the Query and Response. Its created by the Host or the one who make the Query, and then its copied by the DNS Responder and send back in the Respond, that way the Client that made the Request know to which Query this Responses is.

The query:



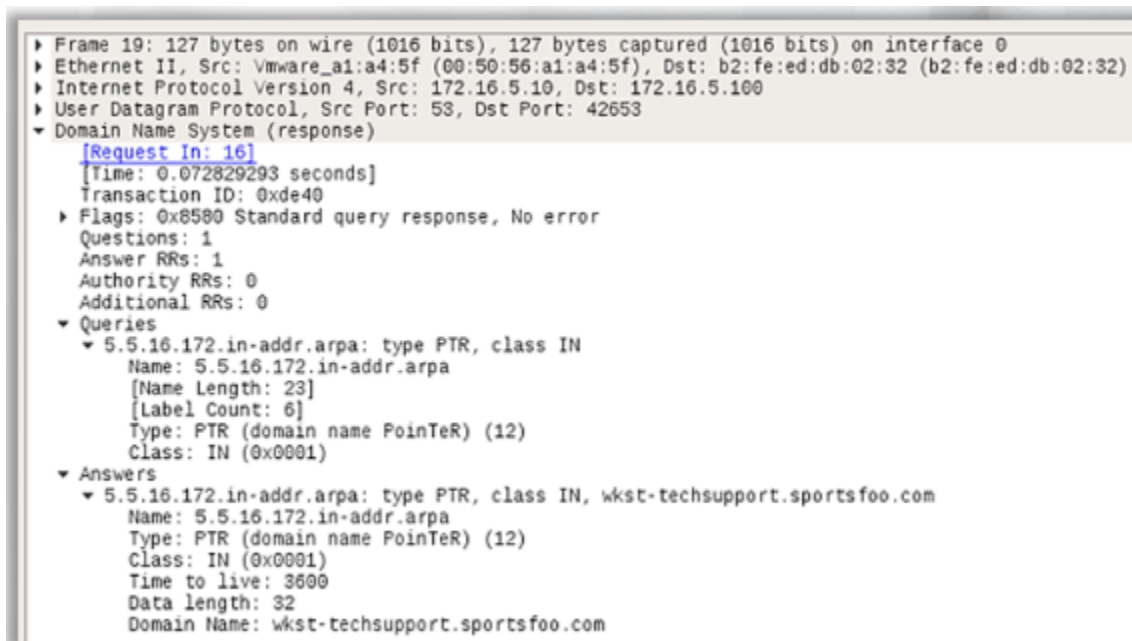
The image shows a Wireshark packet capture of a DNS query. The packet list on the left shows Frame 16: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0. The packet details pane shows the following structure:

- ▶ Ethernet II, Src: b2:fe:ed:db:02:32 (b2:fe:ed:db:02:32), Dst: Vmware_a1:a4:5f (00:50:56:a1:a4:5f)
- ▶ Internet Protocol Version 4, Src: 172.16.5.100, Dst: 172.16.5.10
- ▶ User Datagram Protocol, Src Port: 42653, Dst Port: 53
- ▼ Domain Name System (query)
 - [Response In: 19]
 - Transaction ID: 0xde40
 - ▶ Flags: 0x0100 Standard query
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - ▼ Queries
 - ▼ 5.5.16.172.in-addr.arpa: type PTR, class IN
 - Name: 5.5.16.172.in-addr.arpa
 - [Name Length: 23]

The packet bytes pane shows the raw data of the query, including the transaction ID 0xde40 and the domain name 5.5.16.172.in-addr.arpa.

Here we see the UDP and we see the Destination port is 53 and we also see the 0xde40 transaction ID. And we see this is the query trying to find the IP for “5.5.16.172.in-addr.arpa domain”.

The response:



The image shows a Wireshark packet capture of a DNS response. The packet list on the left shows Frame 19: 127 bytes on wire (1016 bits), 127 bytes captured (1016 bits) on interface 0. The packet details pane shows the following structure:

- ▶ Ethernet II, Src: Vmware_a1:a4:5f (00:50:56:a1:a4:5f), Dst: b2:fe:ed:db:02:32 (b2:fe:ed:db:02:32)
- ▶ Internet Protocol Version 4, Src: 172.16.5.10, Dst: 172.16.5.100
- ▶ User Datagram Protocol, Src Port: 53, Dst Port: 42653
- ▼ Domain Name System (response)
 - [Request In: 16]
 - [Time: 0.072829293 seconds]
 - Transaction ID: 0xde40
 - ▶ Flags: 0x8500 Standard query response, No error
 - Questions: 1
 - Answer RRs: 1
 - Authority RRs: 0
 - Additional RRs: 0
 - ▼ Queries
 - ▼ 5.5.16.172.in-addr.arpa: type PTR, class IN
 - Name: 5.5.16.172.in-addr.arpa
 - [Name Length: 23]
 - [Label Count: 6]
 - Type: PTR (domain name PointeR) (12)
 - Class: IN (0x0001)
 - ▼ Answers
 - ▼ 5.5.16.172.in-addr.arpa: type PTR, class IN, wkst-techsupport.sportsfoo.com
 - Name: 5.5.16.172.in-addr.arpa
 - Type: PTR (domain name PointeR) (12)
 - Class: IN (0x0001)
 - Time to live: 3600
 - Data length: 32
 - Domain Name: wkst-techsupport.sportsfoo.com

Here are some other packets showing DNS queries and Responses:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.170.8	192.168.170.20	DNS	70	Standard query 0x1032 TXT google.com
2	0.000530	192.168.170.20	192.168.170.8	DNS	98	Standard query response 0x1032 TXT google.com TXT
3	4.000522	192.168.170.8	192.168.170.20	DNS	70	Standard query 0xf76f MX google.com
4	4.837355	192.168.170.20	192.168.170.8	DNS	298	Standard query response 0xf76f MX google.com MX 40 smtp4.google.com MX 10 smtp5.google.com.
5	12.817185	192.168.170.8	192.168.170.20	DNS	70	Standard query 0x49a1 LOC google.com
6	12.956209	192.168.170.20	192.168.170.8	DNS	70	Standard query response 0x49a1 LOC google.com
7	20.824827	192.168.170.8	192.168.170.20	DNS	85	Standard query 0x9bbb PTR 104.9.192.66.in-addr.arpa
8	20.825333	192.168.170.20	192.168.170.8	DNS	129	Standard query response 0x9bbb PTR 104.9.192.66.in-addr.arpa PTR 66-192-9-104.gen.twteleco.
9	92.189905	192.168.170.8	192.168.170.20	DNS	74	Standard query 0x75c0 A www.netbsd.org
10	92.238816	192.168.170.20	192.168.170.8	DNS	90	Standard query response 0x75c0 A www.netbsd.org A 204.152.190.12
11	106.965135	192.168.170.8	192.168.170.20	DNS	74	Standard query 0xf0d4 AAAA www.netbsd.org
12	109.202803	192.168.170.20	192.168.170.8	DNS	102	Standard query response 0xf0d4 AAAA www.netbsd.org AAAA 2001:4f8:4:7:2e0:81ff:fe52:9a6b
13	169.827394	192.168.170.8	192.168.170.20	DNS	74	Standard query 0x7f39 AAAA www.netbsd.org
14	169.827781	192.168.170.20	192.168.170.8	DNS	102	Standard query response 0x7f39 AAAA www.netbsd.org AAAA 2001:4f8:4:7:2e0:81ff:fe52:9a6b

↳ Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
↳ Ethernet II, Src: Asustek b1:0c:ad (00:e0:18:b1:0c:ad), Dst: QuantaCo_32:41:8c (00:c0:9f:32:41:8c)
↳ Internet Protocol Version 4, Src: 192.168.170.8, Dst: 192.168.170.20
↳ User Datagram Protocol, Src Port: 32795, Dst Port: 53
↳ Domain Name System (query)

Suspicious Traffic:

As we said before that if we see the DNS traffic on port 53 UDP then its normal DNS but if its 53 TCP then its not normal and we should look into it.

A good example of suspicious DNS traffic would DNS zone transfer.

DNS zone transfer is a way for DNS Servers to replicate DNS database across a group of DNS servers, like a server wants to have the same DNS info as the other DNS server, so it will perform the DNS zone transfer and get all the DNS info from that DNS server.

If an attacker manages to obtain a copy of the entire DNS zone for a domain, they may obtain a complete listing of all hosts in that zone, mean like if we have an internal DNS server that provide info about names, an attacker performing the DNS zone transfer will be able to see all the hosts that are in that DNS server, like if there is a hidden HTTP server that only admins know about it and solve to its domain it uses that internal DNS server, an attacker can get details about that HTTP domain server with DNS zone transfer.

60	82.399428543	172.16.5.100	172.16.5.10	DNS	110	Standard query 0xfc66 AXFR sportsfoo.com OPT
61	82.434731625	172.16.5.10	172.16.5.100	DNS	172	Standard query response 0xfc66 AXFR sportsfoo.com SOA els-winsenr2003.sportsfoo.com OPT
63	82.469319332	172.16.5.10	172.16.5.100	DNS	598	Standard query response 0xfc66 AXFR sportsfoo.com SOA els-winsenr2003.sportsfoo.com

Here we see 3 packets and all packets have the same transaction ID and we see the AXFR which let us know this is DNS query is to perform the DNS Zone transfer.

Lets look at the packet 60, which we saw it was DNS zoon transfer:

```
▶ Frame 60: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
▶ Ethernet II, Src: b2:fe:ed:db:02:32 (b2:fe:ed:db:02:32), Dst: Vmware_a1:a4:5f (00:50:56:a1:a4:5f)
▶ Internet Protocol Version 4, Src: 172.16.5.100, Dst: 172.16.5.10
▶ Transmission Control Protocol, Src Port: 58595, Dst Port: 53, Seq: 1, Ack: 1, Len: 44
▼ Domain Name System (query)
  [Response In: 63]
  Length: 42
  Transaction ID: 0xfc66
  ▶ Flags: 0x0020 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 1
  ▼ Queries
    ▼ sportsfoo.com: type AXFR, class IN
      Name: sportsfoo.com
      [Name Length: 13]
      [Label Count: 2]
      Type: AXFR (transfer of an entire zone) (252)
      Class: IN (0x0001)
  ▶ Additional records
```

Here we see the Port is 53 TCP not UDP, and we see the AXFR which is DNS zone transfer which sending all the DNS information when its requested. The TCP is used with UDP when the Response to a DNS query is too large that dosnt fit in the UDP packet so the query was resubmitted, As we know in DNS zone Transfer is to get all the DNS information so it didn't fit in the UDP packet so it used the TCP, that mean the DNS Zoon Transfer happens over TCP/53 port.

The response to this request:

```
▶ Frame 61: 172 bytes on wire (1376 bits), 172 bytes captured (1376 bits) on interface 0
▶ Ethernet II, Src: Vmware_a1:a4:5f (00:50:56:a1:a4:5f), Dst: b2:fe:ed:db:02:32 (b2:fe:ed:db:02:32)
▶ Internet Protocol Version 4, Src: 172.16.5.10, Dst: 172.16.5.100
▶ Transmission Control Protocol, Src Port: 53, Dst Port: 58595, Seq: 1, Ack: 45, Len: 106
▼ Domain Name System (response)
  [Request In: 60]
  [Time: 0.035303082 seconds]
  Length: 104
  Transaction ID: 0xfc66
  ▶ Flags: 0x8080 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 1
  ▼ Queries
    ▼ sportsfoo.com: type AXFR, class IN
      Name: sportsfoo.com
      [Name Length: 13]
      [Label Count: 2]
      Type: AXFR (transfer of an entire zone) (252)
      Class: IN (0x0001)
  ▼ Answers
    ▼ sportsfoo.com: type SOA, class IN, mname els-winsor2003.sportsfoo.com
      Name: sportsfoo.com
      Type: SOA (Start Of a zone of Authority) (6)
      Class: IN (0x0001)
      Time to live: 3600
      Data length: 50
      Primary name server: els-winsor2003.sportsfoo.com
      Responsible authority's mailbox: hostmaster.sportsfoo.com
      Serial Number: 19
      Refresh Interval: 900 (15 minutes)
      Retry Interval: 600 (10 minutes)
      Expire limit: 86400 (1 day)
      Minimum TTL: 3600 (1 hour)
  ▶ Additional records
```

We see in the answer there a lot of info that sent back as response to that query, so since it's a lot of info then its not gonna fit in the UDP packet, so for that the DNS server will ask the Client to resubmit the Query using TCP so the server can send a response to that Query. (why cant the server response with the TCP when we send the query in UDP? We need first the 3 way handshake that's why, we have to initiate the TCP connection first and we are client so we are not listening on any port that the Server can use to connect to us.)

DNS over TCP is suspicious and if its not blocked then it should be monitored.

DNS can be used in many other attacks like Fast Flux DNS, DGA.. and can also be used as C2, to send commands, and it can also be used to exfiltrate data.