BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Documentation for the project

Monitoring DNS communication

ISA - Network applications and network management

November 15, 2024

Rostyslav Kachan, xkacha02

Contents

1 Description

This project's task is to implement the **dns-monitor** program, which will monitor DNS communication on the selected interface or process DNS messages from an existing communication record in the PCAP format. The tool will process DNS log messages and extract information from them. In addition, the tool will be able to find out what domain names appeared in DNS messages. The third functionality is to search for translations of domain names to IPv4/6 addresses.

2 Introduction to the problem

The Domain Name System (DNS) is a fundamental part of internet infrastructure that translates human-readable domain names (like example.com) into IP addresses that computers use to identify each other on networks. DNS operates in a hierarchical manner, enabling efficient domain lookup across distributed networks.
Key Components of DNS:

- DNS Resolver (Client): Sends requests to DNS servers to resolve domain names into IP addresses.
- DNS Server: Responds to queries with the requested domain's IP address or with referrals to other DNS servers.

DNS Packet Types:

- Query: Sent by the client to request domain information.
- Response: Sent by the server with the result, typically an IP address.

DNS Record Types:

- **A**: IPv4 address for a domain.
- **AAAA**: IPv6 address for a domain.
- **NS**: Name servers for a domain, indicating the authoritative DNS servers.
- **MX**: Mail exchange server for email routing.
- **SOA**: Start of Authority record, providing essential domain information (e.g., primary DNS server, domain serial number).
- **CNAME**: Canonical name for domain aliases, mapping an alias to the true or canonical domain name.
- **SRV**: Service locator, specifying the location (hostname and port) of servers for specific services.

Other record types are not supported in this project.

3 Project Configuration

The **dns-monitor** program is implemented in the C++ programming language, using the C++17 standard. The application is divided into source files. Header files hold system libraries, such as **libpcap** (for packet capture) and others, which are used in the code.

4 Implementation

4.1 main.cpp

The **dns-program** starts execution from the file **main.cpp**. Key elements include variable declarations,signal handling, argument parsing, and validation checks. Variable declarations define flags and data structures: **full_mode** enables verbose output, **uniqueDomains** and **DomainToIP** store unique domains and domain-to-IP mappings, **handle** is the capture pointer, **filter** is the packet filter structure, and **domainsfile** and **translationsfile** specify output filenames.Signal handling is configured to capture termination signals (**SIGTERM**, **SIGINT**, **SIGQUIT**). When a signal is received, the program's **signalHandler()** function is called . The arguments are parsed using the **getopt()** function and the necessary variables are initialized.At the end of the file, validations are made to ensure that the user has entered the data correctly and, depending on the **-i** or **-p** option, the necessary functions are called.

4.2 SignalHandler.cpp and SignalHandler.h

The **signalHandler()** function processes termination signals (**SIGINT**, **SIGTERM**, **SIGQUIT**) by saving domain data to specified files, clearing domain-related sets, releasing pcap filter and handle resources, displaying a termination message, and exiting the program with the received signal code. Here, I implement the preservation of unique domain names or domain-to-IP address mapping when analyzing DNS packets via the interface.

4.3 SetupFilter.cpp and SetupFilter.h

Depending on the application launch parameters, the user calls one of two functions: **captureFromInterface()** or **captureFromFile()**. It opens the source for capture with **pcap_open_live** (for interfaces) or **pcap_open_offline** (for files), handles errors, and applies a filter to capture only DNS packets (UDP port 53) using **pcap_compile** and **pcap_setfilter**. If any errors occur, they are logged with detailed messages. The main packet processing loop is managed by **pcap_loop**, which sends each packet to the **packetHandler()** function. After capture, resources are cleaned up with **pcap_freecode** and **pcap_close**, ensuring proper deallocation.

4.4 PacketHandler.cpp and PacketHandler.h

The **packetHandler()** function processes incoming packets by determining the IP version (IPv4 or IPv6), extracting IP and UDP headers to obtain source and destination information, and parsing DNS headers to obtain details such as transaction ID, flags, and the number of questions, answers, authorities, and additional sections. When **full_mode** is enabled, it prints detailed information about the packet and processes each DNS section (questions, answers, etc.) using the **processDNSQuestions()** and **processDNSSections()** functions. In non-verbose mode, the program displays a summary with a timestamp, IP addresses, and number of records in different sections.

4.5 ProcessDNSSections.cpp and ProcessDNSSections.h

Functions **selectRecordType()** and **selectClass()** print the DNS record type and class based on provided codes.
The **processDNSSections()** function processes DNS packet sections - response, authoritative, and additional - by iterating through each record, extracting data such as type, class, TTL, and length,

and and displays the information contained in RDATA. If the **type** of DNS does not match any of the known types, the program displays the message `UNKNOWN type of record`. This means that the DNS query has a non-standard or unplanned type that is not supported by the program for detailed processing. In such cases, processing of this request is skipped.
The **processDNSQuestions()** function has similar functionality as **processDNSSections()**, but processes only DNS questions.Extract only data domain name, type and class.

4.6 getDomain.cpp and getDomain.h

Function which extracts domain name from DNS packet is **getDomain()**.This function handles all three cases of domain name storage in DNS packets: no compression, compression at the start of the record, and compression in the middle of the domain name.The name of the found domain is stored in a special **hlp** structure (described below) returned by the function.

4.7 SaveFile.cpp and SaveFile.h

addDomain() function add a domain name to **uniqueDomains** set , before removes any trailing dot.
addDomainToIP() has the same functionality but add unique strings to **DomainToIP** set.
Function **saveDomainsIPToFile()** saves either unique domains or domain-to-IP mappings to a specified file based on a flag, creating or opening the file and writing data line by line. If the flag is true, it saves unique domains from the **uniqueDomains set**; if false, it saves domain-to-IP mappings from the **DomainToIP** set.

4.8 Structure 'hlp'

This structure has 3 fields :

- size: An integer representing the size of the domain name or data associated with it.
- name: A string storing the domain name.
- ptr: A pointer (of type unsigned char*) used to indicate a specific position in the DNS packet.

It is used to store domain names, their length, and when processing the name in order to properly move through the DNS packet.

5 Usage

The program can be compiled using the **make** command. The executable file is called **dns-monitor**. To run the client, use:

```
./dns-monitor
```

Necessary command line options(have to choose only one):

- `-i <interface>` - name of the interface on which the program will listen.
- `-p <pcapfile>` - name of the PCAP file that the program will process.

Optional

- `-v` - verbose mode: complete listing of DNS message details.
- `-d <domainsfile>` - the name of the domain name file.
- `-t <translationsfile>` - the name of the domain name to IP translation file.
- `-h` - prints program help output and exits.

6 Testing

This topic would describe application testing and validation. During the testing process, I used such tools as **Wireshark** application to handle and analyze transmitted DNS packets. I tested the program with the **-p** flag both locally and on university servers such as **merlin.fit.vutbr.cz** and **eva.fit.vutbr.cz**. I added the output only from the **merlin.fit.vutbr.cz** server to avoid piling up the documentation. The program with the **-i** flag cannot be tested on these servers due to limited access rights, so I tested it in a local environment. Most of the files (A_test.pcap, AAAA_test.pcap, NS_test.pcap, SOA_test.pcap) for testing were created using the **dig** command, which allows you to generate DNS queries of a certain type. The other files are packets captured from Internet network traffic using **Wireshark** and were used as additional examples for testing. This allowed us to check how the program handles real DNS queries from the network, providing more complete testing in different conditions.

(Example) Command to generate DNS query(type A):

```
dig example.com A
```

6.1 DNS A Record Type (non-verbose)

Command:

```
xkacha02@merlin:˜/ISA$ ./dns-monitor -p A_test.pcap
```

Output:

```
2024-11-11 22:15:18 10.0.2.15 -> 10.0.0.138 (Q 1/0/0/1)
2024-11-11 22:15:18 10.0.0.138 -> 10.0.2.15 (R 1/1/0/1)
```

6.2 DNS A Record Type (verbose)

Command:

```
xkacha02@merlin: ˜/ISA$ ./dns-monitor -p A_test.pcap -v
```

Output:

```
Timestamp: 2024-11-11 22:15:18
SrcIP: 10.0.2.15
DstIP: 10.0.0.138
SrcPort: UDP/57399
DstPort: UDP/53
Identifier: 0x8C62
```

```
Flags: QR=0, OPCODE=0, AA=0, TC=0, RD=1, RA=0, AD=0, CD=0, RCODE=0

[Question Section]
youtube.com. IN A

[Additional Section]
UNKNOWN type of record
====================
Timestamp: 2024-11-11 22:15:18
SrcIP: 10.0.0.138
DstIP: 10.0.2.15
SrcPort: UDP/53
DstPort: UDP/57399
Identifier: 0x8C62
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
youtube.com. IN A

[Answer Section]
youtube.com. 34 IN A 142.251.36.78

[Additional Section]
UNKNOWN type of record
====================
```

6.3 DNS AAAA Record Type (verbose)

Command:

```
xkacha02@merlin: ˜/ISA$ ./dns-monitor -p AAAA_test.pcap -v
```

Output:

```
Timestamp: 2024-11-11 23:18:14
SrcIP: 10.0.2.15
DstIP: 10.0.0.138
SrcPort: UDP/38780
DstPort: UDP/53
Identifier: 0xC8B4
Flags: QR=0, OPCODE=0, AA=0, TC=0, RD=1, RA=0, AD=0, CD=0, RCODE=0

[Question Section]
example.com. IN AAAA

[Additional Section]
```

```
UNKNOWN type of record
====================
Timestamp: 2024-11-11 23:18:14
SrcIP: 10.0.0.138
DstIP: 10.0.2.15
SrcPort: UDP/53
DstPort: UDP/38780
Identifier: 0xC8B4
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
example.com. IN AAAA

[Answer Section]
example.com. 2928 IN AAAA 2606:2800:21f:cb07:6820:80da:af6b:8b2c

[Additional Section]
UNKNOWN type of record
====================
```

6.4 DNS NS Record Type (verbose)

Command:

```
xkacha02@merlin: ˜/ISA$ ./dns-monitor -p NS_test.pcap -v
```

Output:

```
Timestamp: 2024-11-11 23:21:58
SrcIP: 10.0.2.15
DstIP: 10.0.0.138
SrcPort: UDP/46646
DstPort: UDP/53
Identifier: 0xDECE
Flags: QR=0, OPCODE=0, AA=0, TC=0, RD=1, RA=0, AD=0, CD=0, RCODE=0

[Question Section]
example.com. IN NS

[Additional Section]
UNKNOWN type of record
====================
Timestamp: 2024-11-11 23:21:58
SrcIP: 10.0.0.138
DstIP: 10.0.2.15
```

```
SrcPort: UDP/53
DstPort: UDP/46646
Identifier: 0xDECE
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
example.com. IN NS

[Answer Section]
example.com. 86400 IN NS b.iana-servers.net.
example.com. 86400 IN NS a.iana-servers.net.

[Additional Section]
a.iana-servers.net. 1045 IN A 199.43.135.53
a.iana-servers.net. 393 IN AAAA 2001:500:8f::53
b.iana-servers.net. 1624 IN A 199.43.133.53
b.iana-servers.net. 1536 IN AAAA 2001:500:8d::53
UNKNOWN type of record
====================
```

6.5 DNS MX Record Type (verbose)

Command:

```
xkacha02@merlin: ˜/ISA$ ./dns-monitor -p MX_test.pcap -v
```

Output:

```
Timestamp: 1999-03-11 14:45:08
SrcIP: 3ffe:501:4819::42
DstIP: 3ffe:507:0:1:200:86ff:fe05:80da
SrcPort: UDP/53
DstPort: UDP/2397
Identifier: 0x6
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
www.yahoo.com. IN MX

[Answer Section]
www.yahoo.com. 796 IN MX 0 mr1.yahoo.com.

[Authority Section]
yahoo.com. 172696 IN NS ns1.yahoo.com.
yahoo.com. 172696 IN NS ns2.dca.yahoo.com.
yahoo.com. 172696 IN NS ns.europe.yahoo.com.
yahoo.com. 172696 IN NS ns5.dcx.yahoo.com.

[Additional Section]
mr1.yahoo.com. 796 IN A 206.251.17.77
ns5.dcx.yahoo.com. 172695 IN A 216.32.74.10
ns.europe.yahoo.com. 172695 IN A 195.67.49.25
ns2.dca.yahoo.com. 172695 IN A 209.143.200.34
ns1.yahoo.com. 172695 IN A 204.71.200.33
====================
```

6.6 DNS SOA Record Type (verbose)

Command:

```
xkacha02@merlin: ~/ISA$ ./dns-monitor -p SOA_test.pcap -v
```

Output:

```
Timestamp: 2024-11-11 23:23:05
SrcIP: 10.0.2.15
DstIP: 10.0.0.138
SrcPort: UDP/60655
DstPort: UDP/53
Identifier: 0xEDF9
Flags: QR=0, OPCODE=0, AA=0, TC=0, RD=1, RA=0, AD=0, CD=0, RCODE=0

[Question Section]
example.com. IN SOA

[Additional Section]
UNKNOWN type of record
====================
Timestamp: 2024-11-11 23:23:05
SrcIP: 10.0.0.138
DstIP: 10.0.2.15
SrcPort: UDP/53
DstPort: UDP/60655
Identifier: 0xEDF9
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
example.com. IN SOA
```

```
[Answer Section]
example.com. 3600 IN SOA ns.icann.org. noc.dns.icann.org. 2024081452 7200
3600 1209600 3600

[Additional Section]
UNKNOWN type of record
====================
```

6.7 DNS CNAME Record Type (verbose)

Command:

```
xkacha02@merlin: ~/ISA$ ./dns-monitor -p CNAME_test.pcap -v
```

Output:

```
Timestamp: 2024-11-11 23:31:41
SrcIP: 10.0.0.138
DstIP: 10.0.2.15
SrcPort: UDP/53
DstPort: UDP/55212
Identifier: 0xC93C
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
www.paypalobjects.com. IN A

[Answer Section]
www.paypalobjects.com. 2715 IN CNAME ppo.glb.paypal.com.
ppo.glb.paypal.com. 16 IN CNAME cs1150.wpc.betacdn.net.
cs1150.wpc.betacdn.net. 1754 IN A 192.229.221.25

[Additional Section]
UNKNOWN type of record
====================
```

6.8 DNS SRV Record Type (verbose)

Command:

```
xkacha02@merlin: ~/ISA$ ./dns-monitor -p SRV_test.pcap -v
```

Output:

```
Timestamp: 2024-11-11 23:53:58
SrcIP: 10.0.2.15
DstIP: 8.8.8.8
SrcPort: UDP/42913
DstPort: UDP/53
Identifier: 0x78F0
Flags: QR=0, OPCODE=0, AA=0, TC=0, RD=1, RA=0, AD=1, CD=0, RCODE=0

[Question Section]
_sip._tls.teams.microsoft.com. IN SRV

[Additional Section]
UNKNOWN type of record
====================
Timestamp: 2024-11-11 23:53:58
SrcIP: 8.8.8.8
DstIP: 10.0.2.15
SrcPort: UDP/53
DstPort: UDP/42913
Identifier: 0x78F0
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=3

[Question Section]
_sip._tls.teams.microsoft.com. IN SRV

[Authority Section]
microsoft.com. 300 IN SOA ns1-39.azure-dns.com.
azuredns-hostmaster.microsoft.com. 1 3600 300 2419200 300

[Additional Section]
UNKNOWN type of record
====================
```

6.9 DNS MX Record Type with -d flag

Command:

```
xkacha02@merlin: ~/ISA$ ./dns-monitor -p MX_test.pcap -d domain.txt
```

Output:

```
1999-03-11 14:45:08 3ffe:501:4819::42 -> 3ffe:507:0:1:200:86ff:fe05:80da
(R 1/1/4/5)
```

”domain.txt” output:

```
mr1.yahoo.com
ns.europe.yahoo.com
ns1.yahoo.com
ns2.dca.yahoo.com
ns5.dcx.yahoo.com
www.yahoo.com
yahoo.com
```

6.10 DNS MX Record Type with -t flag

Command:

```
xkacha02@merlin: ~/ISA$ ./dns-monitor -p MX_test.pcap -t translation.txt
```

Output:

```
1999-03-11 14:45:08 3ffe:501:4819::42 -> 3ffe:507:0:1:200:86ff:fe05:80da
(R 1/1/4/5)
```

”translation.txt” output:

```
mr1.yahoo.com 206.251.17.77
ns.europe.yahoo.com 195.67.49.25
ns1.yahoo.com 204.71.200.33
ns2.dca.yahoo.com 209.143.200.34
ns5.dcx.yahoo.com 216.32.74.10
```

6.11 Capture from interface (-i)

First command:

```
sudo ./dns-monitor -i enp0s3 -v -d domainsInterface.txt
-t translationInterface.txt
```

Second command:

```
dig youtube.com AAAA
```

Output:

```
Timestamp: 2024-11-12 15:02:22
SrcIP: 10.0.2.15
DstIP: 10.0.0.138
SrcPort: UDP/48099
DstPort: UDP/53
Identifier: 0x7827
```

```
Flags: QR=0, OPCODE=0, AA=0, TC=0, RD=1, RA=0, AD=0, CD=0, RCODE=0

[Question Section]
youtube.com. IN AAAA

[Additional Section]
UNKNOWN type of record
====================
Timestamp: 2024-11-12 15:02:22
SrcIP: 10.0.0.138
DstIP: 10.0.2.15
SrcPort: UDP/53
DstPort: UDP/48099
Identifier: 0x7827
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
youtube.com. IN AAAA

[Answer Section]
youtube.com. 126 IN AAAA 2a00:1450:4014:80f::200e

[Additional Section]
UNKNOWN type of record
====================
ˆCCatch SIGINT, SIGTERM, SIGQUIT
```

”domainsInterface.txt” output:

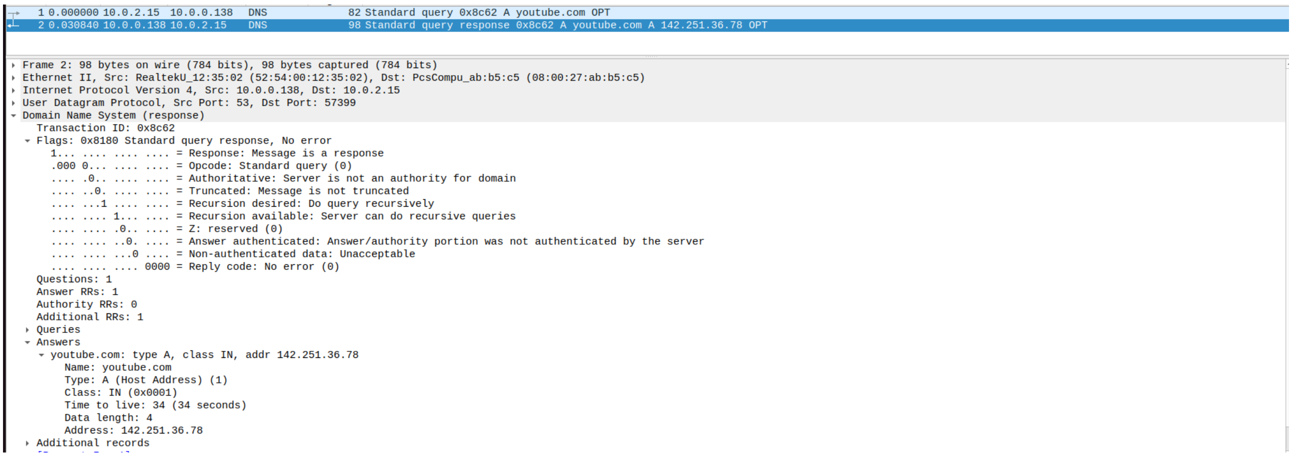
```
youtube.com
```

”translationInterface.txt” output:

```
youtube.com 2a00:1450:4014:80f::200e
```

6.12 Wireshark Screenshots

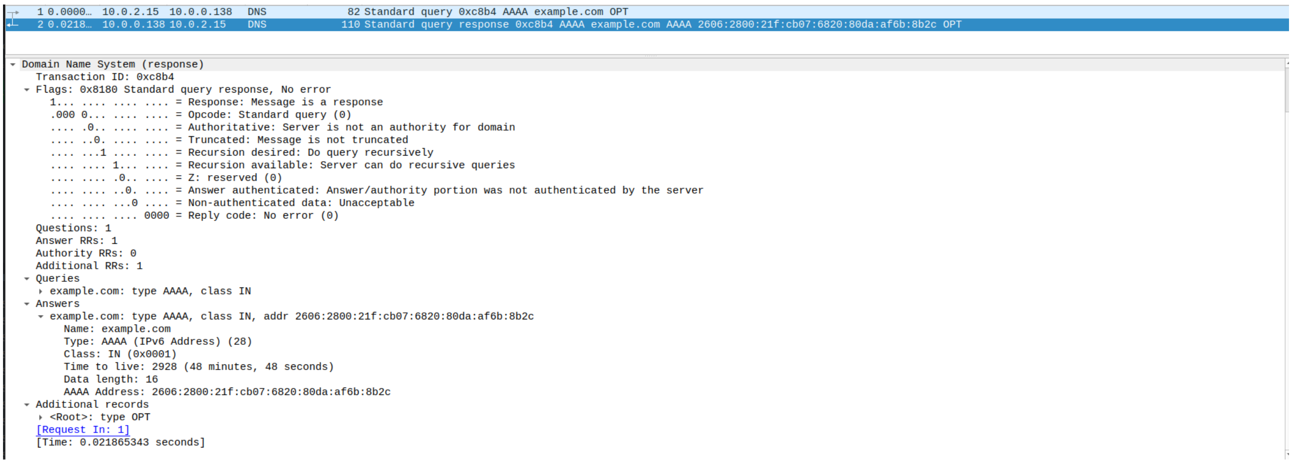
This section contains screenshots of packages with different types of DNS records in Wireshark that were used for testing.



```
1 0.000000 10.0.2.15  10.0.0.138  DNS        82 Standard query 0x8c62 A youtube.com OPT
2 0.030840 10.0.0.138 10.0.2.15   DNS        98 Standard query response 0x8c62 A youtube.com A 142.251.36.78 OPT

▸ Frame 2: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
▸ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_ab:b5:c5 (08:00:27:ab:b5:c5)
▸ Internet Protocol Version 4, Src: 10.0.0.138, Dst: 10.0.2.15
▸ User Datagram Protocol, Src Port: 53, Dst Port: 57399
▾ Domain Name System (response)
    Transaction ID: 0x8c62
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  ▸ Queries
  ▾ Answers
    ▾ youtube.com: type A, class IN, addr 142.251.36.78
        Name: youtube.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 34 (34 seconds)
        Data length: 4
        Address: 142.251.36.78
  ▸ Additional records
```

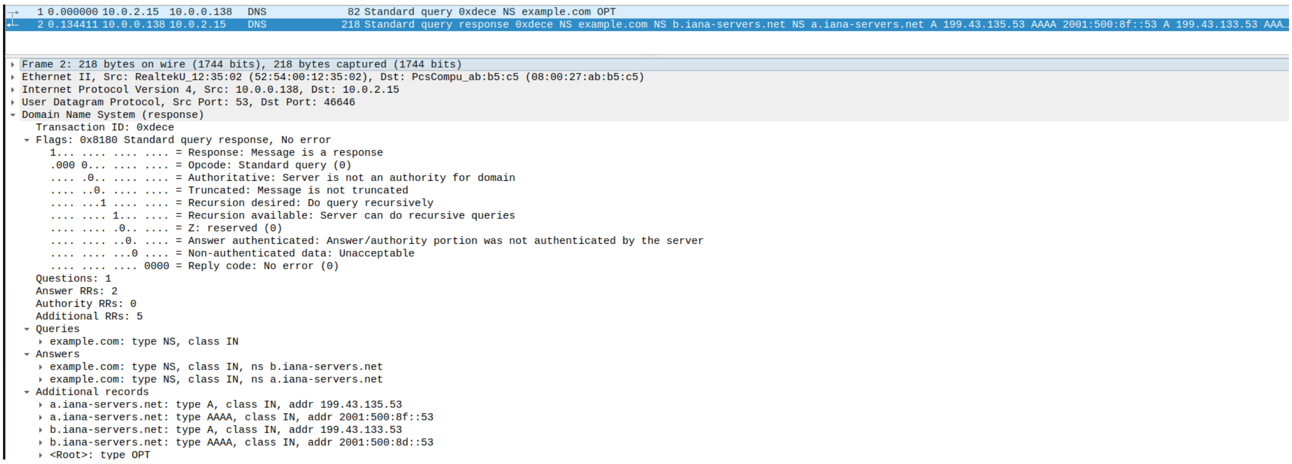
Figure 1: Screenshot of **A_test.pcap**.



```
1 0.0000…  10.0.2.15  10.0.0.138  DNS        82 Standard query 0xc8b4 AAAA example.com OPT
2 0.0218…  10.0.0.138 10.0.2.15   DNS       110 Standard query response 0xc8b4 AAAA example.com AAAA 2606:2800:21f:cb07:6820:80da:af6b:8b2c OPT

▾ Domain Name System (response)
    Transaction ID: 0xc8b4
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▸ example.com: type AAAA, class IN
  ▾ Answers
    ▾ example.com: type AAAA, class IN, addr 2606:2800:21f:cb07:6820:80da:af6b:8b2c
        Name: example.com
        Type: AAAA (IPv6 Address) (28)
        Class: IN (0x0001)
        Time to live: 2928 (48 minutes, 48 seconds)
        Data length: 16
        AAAA Address: 2606:2800:21f:cb07:6820:80da:af6b:8b2c
  ▾ Additional records
    ▸ <Root>: type OPT
    [Request In: 1]
    [Time: 0.021865343 seconds]
```

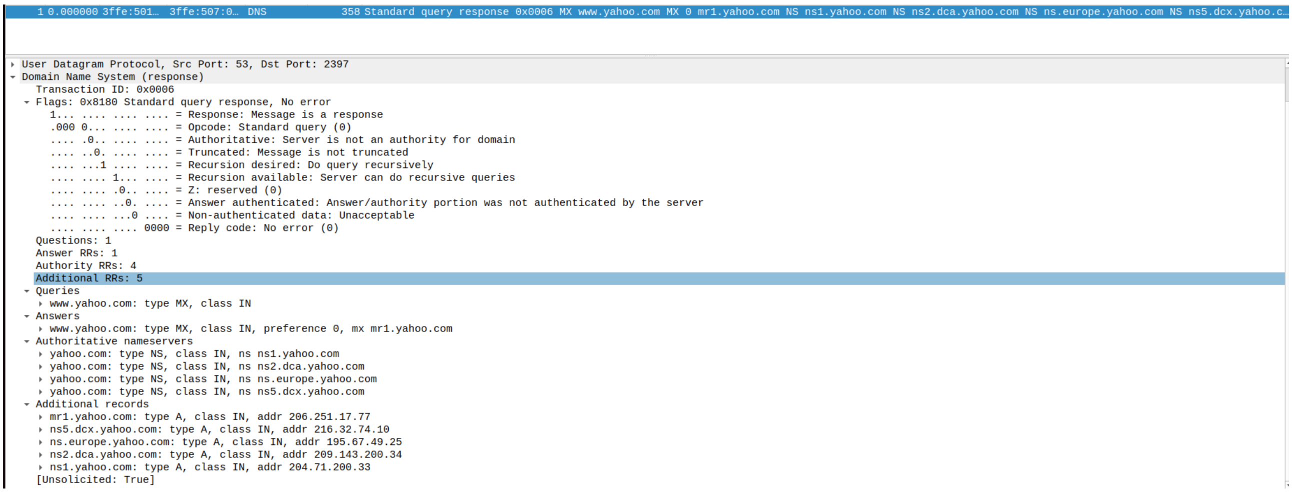
Figure 2: Screenshot of **AAAA_test.pcap**.



```
1 0.000000 10.0.2.15  10.0.0.138  DNS        82 Standard query 0xdece NS example.com OPT
2 0.134411 10.0.0.138 10.0.2.15   DNS       218 Standard query response 0xdece NS example.com NS b.iana-servers.net NS a.iana-servers.net A 199.43.135.53 AAAA 2001:500:8f::53 A 199.43.133.53 AAA…

▸ Frame 2: 218 bytes on wire (1744 bits), 218 bytes captured (1744 bits)
▸ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_ab:b5:c5 (08:00:27:ab:b5:c5)
▸ Internet Protocol Version 4, Src: 10.0.0.138, Dst: 10.0.2.15
▸ User Datagram Protocol, Src Port: 53, Dst Port: 46646
▾ Domain Name System (response)
    Transaction ID: 0xdece
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 5
  ▾ Queries
    ▸ example.com: type NS, class IN
  ▾ Answers
    ▸ example.com: type NS, class IN, ns b.iana-servers.net
    ▸ example.com: type NS, class IN, ns a.iana-servers.net
  ▾ Additional records
    ▸ a.iana-servers.net: type A, class IN, addr 199.43.135.53
    ▸ a.iana-servers.net: type AAAA, class IN, addr 2001:500:8f::53
    ▸ b.iana-servers.net: type A, class IN, addr 199.43.133.53
    ▸ b.iana-servers.net: type AAAA, class IN, addr 2001:500:8d::53
    ▸ <Root>: type OPT
```

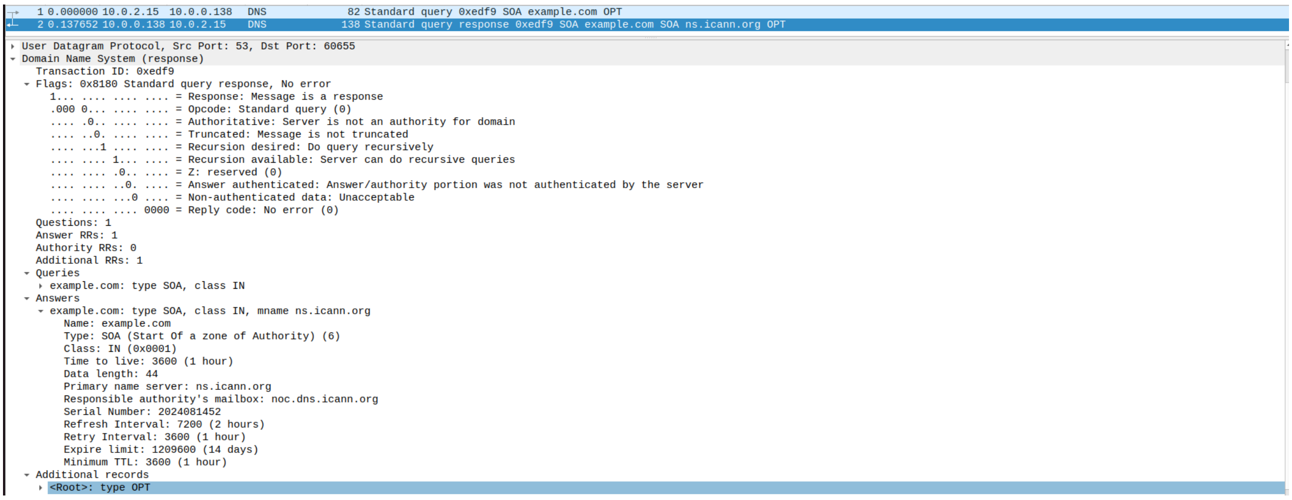
Figure 3: Screenshot of **NS_test.pcap**.



```
1 0.000000 3ffe:501…  3ffe:507:0…  DNS        358 Standard query response 0x0006 MX www.yahoo.com MX 0 mr1.yahoo.com NS ns1.yahoo.com NS ns2.dca.yahoo.com NS ns.europe.yahoo.com NS ns5.dcx.yahoo.c…

▸ User Datagram Protocol, Src Port: 53, Dst Port: 2397
▾ Domain Name System (response)
    Transaction ID: 0x0006
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 4
    Additional RRs: 5
  ▾ Queries
    ▸ www.yahoo.com: type MX, class IN
  ▾ Answers
    ▸ www.yahoo.com: type MX, class IN, preference 0, mx mr1.yahoo.com
  ▾ Authoritative nameservers
    ▸ yahoo.com: type NS, class IN, ns ns1.yahoo.com
    ▸ yahoo.com: type NS, class IN, ns ns2.dca.yahoo.com
    ▸ yahoo.com: type NS, class IN, ns ns.europe.yahoo.com
    ▸ yahoo.com: type NS, class IN, ns ns5.dcx.yahoo.com
  ▾ Additional records
    ▸ mr1.yahoo.com: type A, class IN, addr 206.251.17.77
    ▸ ns5.dcx.yahoo.com: type A, class IN, addr 216.32.74.10
    ▸ ns.europe.yahoo.com: type A, class IN, addr 195.67.49.25
    ▸ ns2.dca.yahoo.com: type A, class IN, addr 209.143.200.34
    ▸ ns1.yahoo.com: type A, class IN, addr 204.71.200.33
    [Unsolicited: True]
```

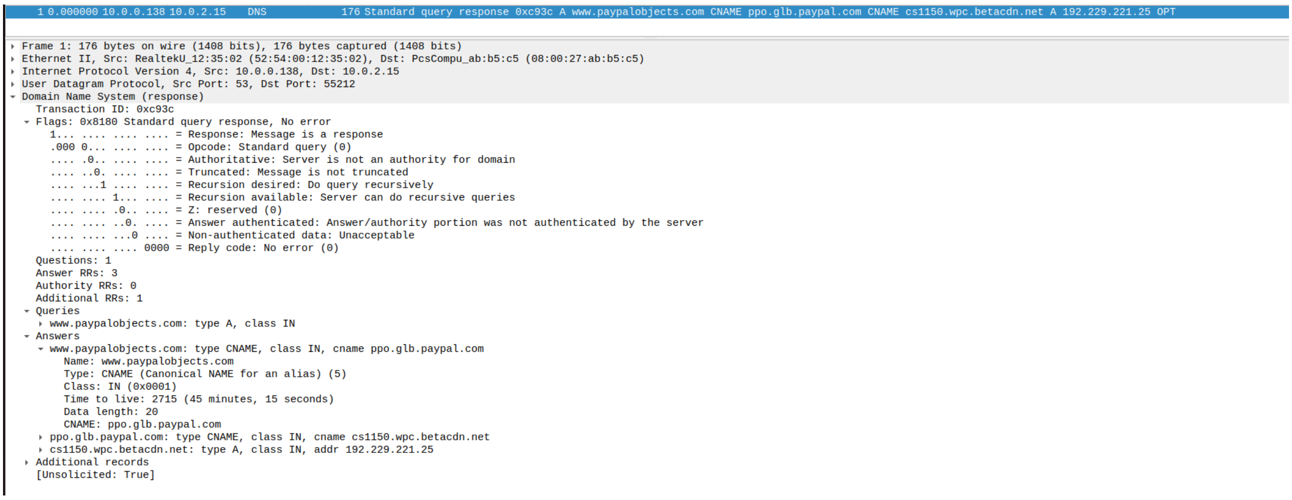
Figure 4: Screenshot of **MX_test.pcap**.



```
1 0.000000 10.0.2.15  10.0.0.138   DNS         82 Standard query 0xedf9 SOA example.com OPT
2 0.137652 10.0.0.138 10.0.2.15    DNS        138 Standard query response 0xedf9 SOA example.com SOA ns.icann.org OPT

▸ User Datagram Protocol, Src Port: 53, Dst Port: 60655
▾ Domain Name System (response)
    Transaction ID: 0xedf9
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▸ example.com: type SOA, class IN
  ▾ Answers
    ▾ example.com: type SOA, class IN, mname ns.icann.org
        Name: example.com
        Type: SOA (Start Of a zone of Authority) (6)
        Class: IN (0x0001)
        Time to live: 3600 (1 hour)
        Data length: 44
        Primary name server: ns.icann.org
        Responsible authority's mailbox: noc.dns.icann.org
        Serial Number: 2024081452
        Refresh Interval: 7200 (2 hours)
        Retry Interval: 3600 (1 hour)
        Expire limit: 1209600 (14 days)
        Minimum TTL: 3600 (1 hour)
  ▾ Additional records
    ▸ <Root>: type OPT
```

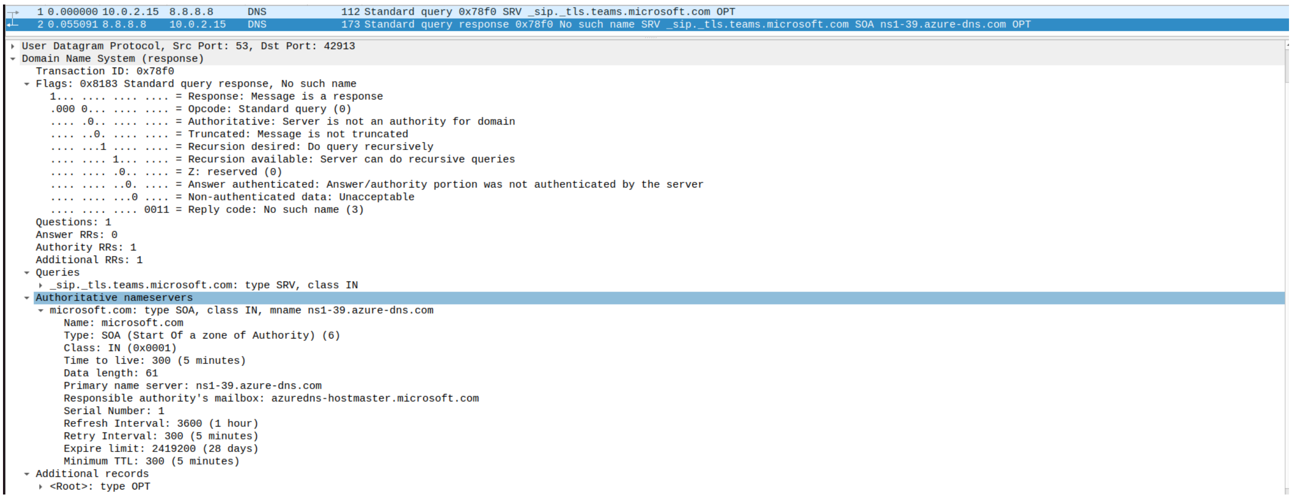
Figure 5: Screenshot of **SOA_test.pcap**.



```
1 0.000000 10.0.0.138 10.0.2.15    DNS        176 Standard query response 0xc93c A www.paypalobjects.com CNAME ppo.glb.paypal.com CNAME cs1150.wpc.betacdn.net A 192.229.221.25 OPT

▸ Frame 1: 176 bytes on wire (1408 bits), 176 bytes captured (1408 bits)
▸ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_ab:b5:c5 (08:00:27:ab:b5:c5)
▸ Internet Protocol Version 4, Src: 10.0.0.138, Dst: 10.0.2.15
▸ User Datagram Protocol, Src Port: 53, Dst Port: 55212
▾ Domain Name System (response)
    Transaction ID: 0xc93c
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 3
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▸ www.paypalobjects.com: type A, class IN
  ▾ Answers
    ▾ www.paypalobjects.com: type CNAME, class IN, cname ppo.glb.paypal.com
        Name: www.paypalobjects.com
        Type: CNAME (Canonical NAME for an alias) (5)
        Class: IN (0x0001)
        Time to live: 2715 (45 minutes, 15 seconds)
        Data length: 20
        CNAME: ppo.glb.paypal.com
    ▸ ppo.glb.paypal.com: type CNAME, class IN, cname cs1150.wpc.betacdn.net
    ▸ cs1150.wpc.betacdn.net: type A, class IN, addr 192.229.221.25
  ▸ Additional records
    [Unsolicited: True]
```

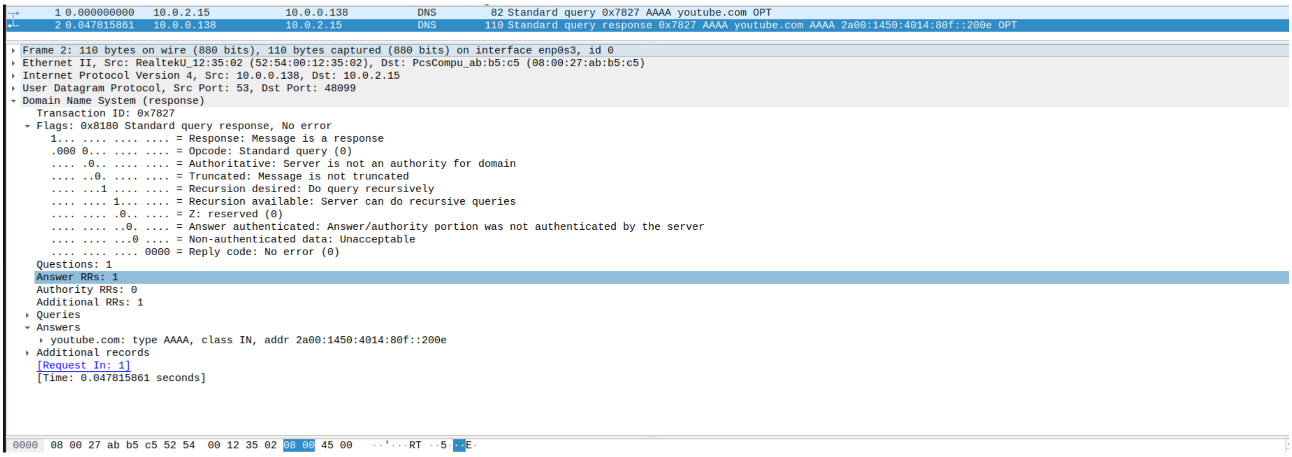
Figure 6: Screenshot of **CNAME_test.pcap**.



```
    1 0.000000 10.0.2.15  8.8.8.8     DNS          112 Standard query 0x78f0 SRV _sip._tls.teams.microsoft.com OPT
    2 0.055091 8.8.8.8    10.0.2.15   DNS          173 Standard query response 0x78f0 No such name SRV _sip._tls.teams.microsoft.com SOA ns1-39.azure-dns.com OPT

▸ User Datagram Protocol, Src Port: 53, Dst Port: 42913
▾ Domain Name System (response)
    Transaction ID: 0x78f0
  ▾ Flags: 0x8183 Standard query response, No such name
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0011 = Reply code: No such name (3)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 1
  ▾ Queries
    ▸ _sip._tls.teams.microsoft.com: type SRV, class IN
  ▾ Authoritative nameservers
    ▾ microsoft.com: type SOA, class IN, mname ns1-39.azure-dns.com
        Name: microsoft.com
        Type: SOA (Start Of a zone of Authority) (6)
        Class: IN (0x0001)
        Time to live: 300 (5 minutes)
        Data length: 61
        Primary name server: ns1-39.azure-dns.com
        Responsible authority's mailbox: azuredns-hostmaster.microsoft.com
        Serial Number: 1
        Refresh Interval: 3600 (1 hour)
        Retry Interval: 300 (5 minutes)
        Expire limit: 2419200 (28 days)
        Minimum TTL: 300 (5 minutes)
  ▾ Additional records
    ▸ <Root>: type OPT
```

Figure 7: Screenshot of **SRV_test.pcap**.



```
    1 0.000000000   10.0.2.15         10.0.0.138        DNS        82 Standard query 0x7827 AAAA youtube.com OPT
    2 0.047815861   10.0.0.138        10.0.2.15         DNS       110 Standard query response 0x7827 AAAA youtube.com AAAA 2a00:1450:4014:80f::200e OPT

▸ Frame 2: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_ab:b5:c5 (08:00:27:ab:b5:c5)
▸ Internet Protocol Version 4, Src: 10.0.0.138, Dst: 10.0.2.15
▸ User Datagram Protocol, Src Port: 53, Dst Port: 48099
▾ Domain Name System (response)
    Transaction ID: 0x7827
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  ▸ Queries
  ▾ Answers
    ▸ youtube.com: type AAAA, class IN, addr 2a00:1450:4014:80f::200e
  ▸ Additional records
    [Request In: 1]
    [Time: 0.047815861 seconds]

0000  08 00 27 ab b5 c5 52 54  00 12 35 02 08 00 45 00   ··'···RT ··5···E·
```

Figure 8: Screenshot of packet used for test interface **-i**.

6.13 Testing Summary

The program underwent testing across all packet types , no memory leaks were detected. Tests were conducted both directly on an interface and with PCAP file handling and the results were cross-checked with Wireshark to ensure precise monitoring of DNS communication.

7 Literature

References

[1] P. Mockapetris, *Domain names - Implementation and specification*, RFC 1035, November 1987. Available: `https://datatracker.ietf.org/doc/html/rfc1035`

[2] R. A. Wright, *Domain Name System Security Extensions*, RFC 2065, January 1997. Available: `https://datatracker.ietf.org/doc/html/rfc2065`

[3] R. H. Thayer, *DNS Extensions to Support IP Version 6*, RFC 3596, October 2003. Available: `https://datatracker.ietf.org/doc/html/rfc3596`

[4] *DNS Packet Structure*, Mislove. Available: `https://mislove.org/teaching/cs4700/spring11/handouts/project1-primer.pdf`

[5] *Domain Name System*, Wikipedia. Available: `https://en.wikipedia.org/wiki/Domain_Name_System`

[6] *DNS Fundamentals and Packet Structure*, Nullhardware. Available: `https://www.nullhardware.com/blog/dns-basics/`