CHAPTER 1: TCP HANDSHAKE

Three-Way TCP Handshake is a process used in a TCP/IP network to establish a connection between the server and client. Before the actual data communication process can begin, both the client and server must exchange synchronisation and acknowledgment packets.

The three-way handshake process is designed so that both ends assist you in initiating, negotiating, and separating TCP socket connections at the same time. It enables the simultaneous transfer of multiple TCP socket connections in both directions.

Steps to establish TCP Handshake:

Step 1 (SYN): In the first step, the client wishes to establish a connection with a server, so it sends a segment with SYN (Synchronize Sequence Number) to inform the server that the client intends to initiate communication and with what sequence number it will begin segments.

Step 2 (SYN + ACK): The server responds to the client's request by setting the SYN-ACK signal bits. Acknowledgement (ACK) denotes the response of the segment received, and SYN denotes the sequence number with which the segments are likely to begin.

Step 3 (Acknowledgement): In the final section, the client acknowledges the server's response and they both establish a reliable connection with which they will begin the actual data transfer. Above all steps are mandatory for the TCP Handshake [1].

TCP Handshake can be analysed using Wireshark Tool. TCP packets can be captured using this tool as shown in Figure 1. We can configure the IP addresses and Port Numbers of source host and destination host. We can also confirm that there is proper TCP handshake between the two hosts by checking their respective flag bits set.

From the Figure 2, we can observe that the first TCP packet which is send from source to destination, its SYN bit is set. The SYN and ACK bit should be set for the second TCP packet which is sent from destination host to source host (Figure 3). After source receives second packet, it sends the third and final packet to destination which has ACK bit set. (Figure 4)

After all this verification of above three steps we can say that there is successful TCP handshake is made between the source and destination. Now the source host can start data communication with the destination host. If any of the three steps of TCP handshake fails, it will lead to issues like:

- No successful data transfer between sender host and destination host.
- Packets will be lost before it reaches destination host.
- Hacker can manipulate the flag bits which can lead to denial of service.

After the data communication between source and destination over, there is need for successful teardown of TCP connection is very important otherwise it can lead to misuse of the handshake, traffic can increase.

CHAPTER 2: SSL/TLS Cipher Suites

Cipher suites are sets of instructions that enable secure network connections through Transport Layer Security (TLS), often still referred to as Secure Sockets Layer (SSL). Behind the scenes, these cipher suites provide a set of algorithms and protocols required to secure communications between clients and servers.

To initiate an HTTPS connection, the two parties – the web server and the client – perform an SSL handshake. The handshake process is a fairly complicated one, during which the two parties agree on a mutual cipher suite. The cipher suite is then used to negotiate a secure HTTPS connection. These ciphers are required at various points of the connection to perform authentication, key generation and exchange, and a checksum to ensure integrity. To determine what specific algorithms to use, the client and the web server start by mutually deciding on the cipher suite to be used.

During a connection's handshake, when the client and server exchange information, the web server and browser compare their high priority lists of supported cypher suites to determine compatibility and which cypher suite to use. The web server makes the decision on which cypher suite to use. The agreed-upon cypher suite consists of:

- Key exchange algorithms, such as RSA, DH, ECDH, DHE, ECDHE, or PSK
- Authentication/Digital Signature Algorithm, like RSA, ECDSA, or DSA
- Bulk encryption algorithms, like AES, CHACHA20, Camellia, or ARIA
- Message Authentication Code algorithms, such as SHA-256, and POLY1305

Cipher suites are required because of the variety of servers, operating systems and browsers. There needs to be a way to accommodate all these combinations, so cipher suites come in handy to ensure compatibility [2].

We can analyse SSL cipher suites using Wireshark tool. Figure 5 shows the snapshot of one of the Client Hello packets of SSLv3, Transport layer Security description. We can see from the snapshot that there are 11 cipher suites for the client packet. Whereas Figure 6 shows the snapshot of the Server Hello packet of SSLv3 Transport layer Security description. We can see from the snapshot that there is only one cipher suite for the server packet.

CHAPTER 3: KERBEROS

Kerberos provides a centralised authentication server that authenticates users to servers and servers to users. For client authentication in Kerberos Authentication, a server and a database are used. Kerberos operates as a trusted third-party server known as the Key Distribution Centre (KDC).

The main components of Kerberos are:

- **Authentication Server (AS):**
  The Authentication Server performs the initial authentication and ticket for Ticket Granting Service.
- **Database:**
  The Authentication Server verifies the access rights of users in the database.
- **Ticket Granting Server (TGS):**
  The Ticket Granting Server issues the ticket for the Server [3].

Wireshark tool can be used to analyse Kerberos protocol also. Kerberos protocol is represented as KRB5 in Wireshark.

- Figure 7 shows the Authentication Server Request Information (AS-REQ) It is a message sent to the authentication service is known as an AS-REQ (AS). The AS only exchanges credentials for tickets. These credentials can be anything, but they are typically passwords.
- Figure 8 shows KRB error message.
- Figure 9 shows the Authentication Server Response Information (AS-REP), In the snapshot it is evident that there isn't any password information in there. That's due to the key agreement it does not don't leak the key. Because the KDC knows the password, it generates a response (AS-REP).
- Figure 10 shows the Ticket Granting Server Request Information (TGS-REQ), The TGS-REQ and AS-REQ are nearly identical. In fact, the message structure is the same. The difference is that we include something called pre-auth data, which is that encrypted ticket. The message contains the name of the requested service as well as the ticket granting ticket in the preauth-data.
  When the TGS receives this message, it checks the preauth-data before extracting that TGT. The TGT is encrypted to the krbtgt long term key, which serves as the password. Only the KDC knows the krbtgt password, so it knows it's genuine and originated with it.
- We now have that session key, which only the client (and now TGS) knows. The KDC generates a response (REP) and encrypts the client metadata to the session key rather than the client password. Figure 11 shows the Ticket Granting Server Response Information (TGS-REP) [4].

Kerberos security authentication protocols use cryptography, multiple secret keys, and third-party authorization to create a strong, secure defence. Passwords are not transmitted across networks, and all secret keys are encrypted. But still an unauthenticated remote attacker can impersonate the Kerberos key distribution centre (KDC) and bypass authentication on a vulnerable device.

CHAPTER 4: DNS ANOMALIES

Web browsers communicate using Internet Protocol (IP) addresses. DNS converts domain names to IP addresses so that browsers can access Internet resources. Each Internet-connected device has a unique IP address that other machines can use to locate the device. DNS resolution is the process of converting a hostname (such as www.netsec.com) into a computer-friendly IP address (such as 192.168.1.1). Each device on the Internet is assigned an IP address, and that address is required to locate the appropriate Internet device. When a user requests a webpage, a translation must take place between what the user types into their web browser (example.com) and the machine-friendly address required to locate the example.com webpage [5].

To understand the functionality of DNS, we must understand some basic terminologies first:

- **Resolver:** A DNS client that sends DNS messages to obtain information about the requested domain name space.
- **Recursion:** The action is taken when a DNS server is asked to query on behalf of a DNS resolver.
- **Authoritative Server:** A DNS server that responds to query messages with information stored in Resource Records for a domain namespace stored on the server.
- **Recursive Resolver:** A DNS server that recursively queries for the information asked in the DNS query.
- **Fully Qualified Domain Name (FQDN):** A Fully Qualified Domain Name is the absolute name of a device within the distributed DNS database.
- **RR:** A Resource Record is a format used in DNS messages that is composed of the following fields: NAME, TYPE, CLASS, TTL, RDLENGTH, and RDATA.
- **Zone:** A database that contains information about the domain name space stored on an authoritative server [6].

The DNS system, like many Internet protocols, was not designed with security in mind and has several design limitations. Because of these limitations, as well as technological advancements, DNS servers are vulnerable to a wide range of attacks, including spoofing, amplification, DoS (Denial of Service), and the interception of private personal information. And, because DNS is used in almost all Internet requests, it can be a prime target for attackers. Furthermore, DNS attacks are frequently used in conjunction with other cyberattacks to divert security teams' attention away from the true target. An organisation must be able to quickly mitigate DNS attacks so that it is not overburdened by simultaneous attacks via other vectors [5].

Using Wireshark tool, we can also analyse the DNS Anomalies. To analyse DNS anomalies, we can capture the DNS Packet and check its query and response messages for anomalies.

Figure 12 shows the snapshot for the DNS Standard query description for the website gaia.umass.edu. Figure 13 shows the snapshot for the DNS Standard query response description for the website gaia.umass.edu which has authoritative nameservers description. We can see that in the response message there is no error is set in reply code, means the tested response message is error-free.

REFERENCES

[1] [Online]. Available: https://www.geeksforgeeks.org/tcp-3-way-handshake-process/..

[2] [Online]. Available: https://www.keyfactor.com/blog/cipher-suites-explained/.

[3] [Online]. Available: https://www.geeksforgeeks.org/kerberos/.

[4] [Online]. Available: https://syfuhs.net/a-bit-about-kerberos.

[5] [Online]. Available: https://www.cloudflare.com/learning/dns/what-is-dns/.

[6] [Online]. Available: https://resources.infosecinstitute.com/topic/detection-prevention-dns-anomalies/.

APPENDIX

Chapter 1

Figure 1

Snapshot TCP packets capture

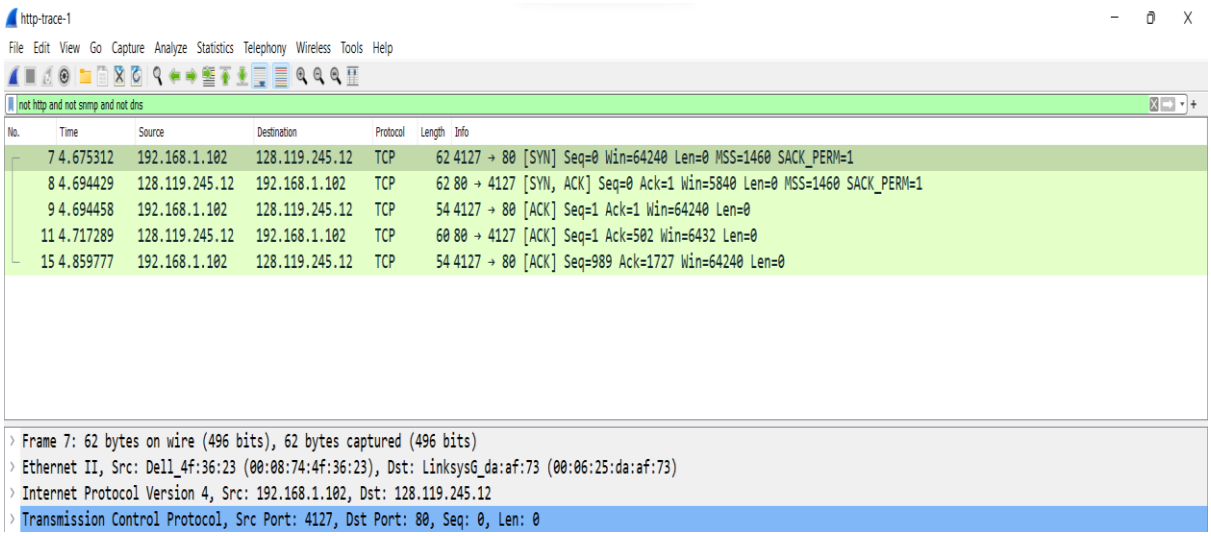


Figure 2

Snapshot of flags for first TCP packet for Handshake

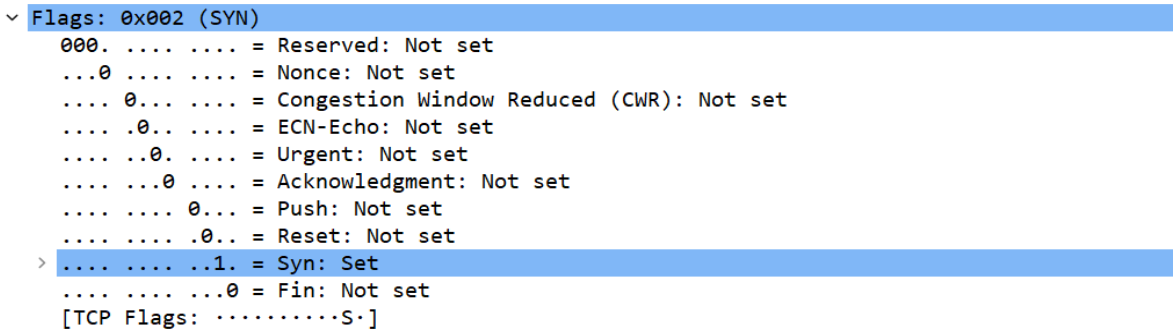


Figure 3

Snapshot of flags for second TCP packet for Handshake

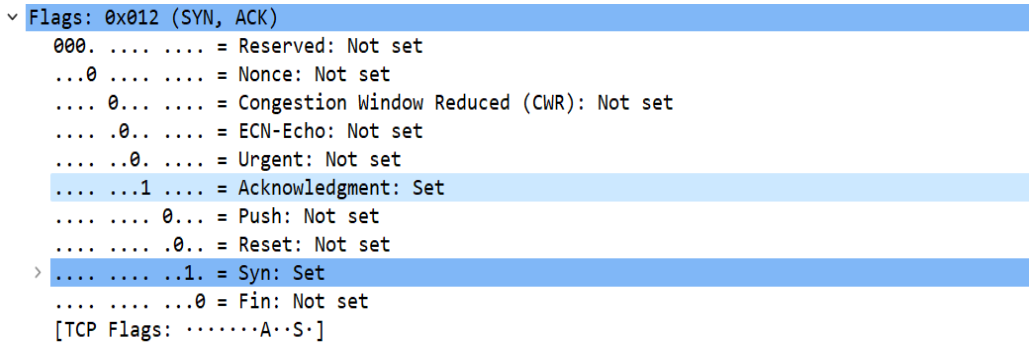
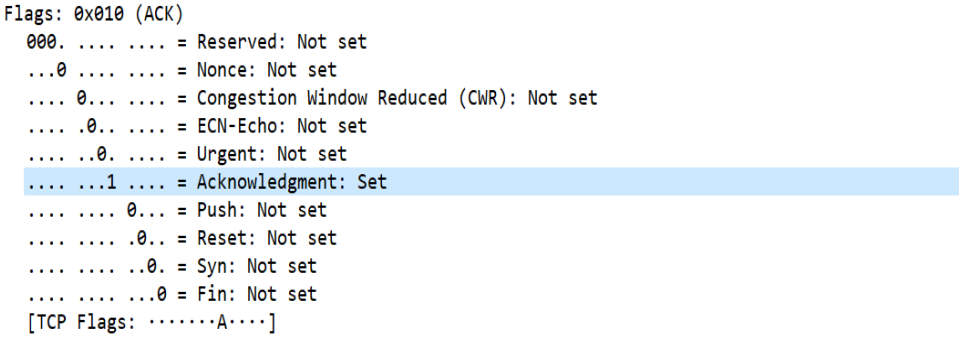


Figure 4

Snapshot of flags for third TCP packet for Handshake



CHAPTER 2

Figure 5

Snapshot of Client Hello Packet TLS Description

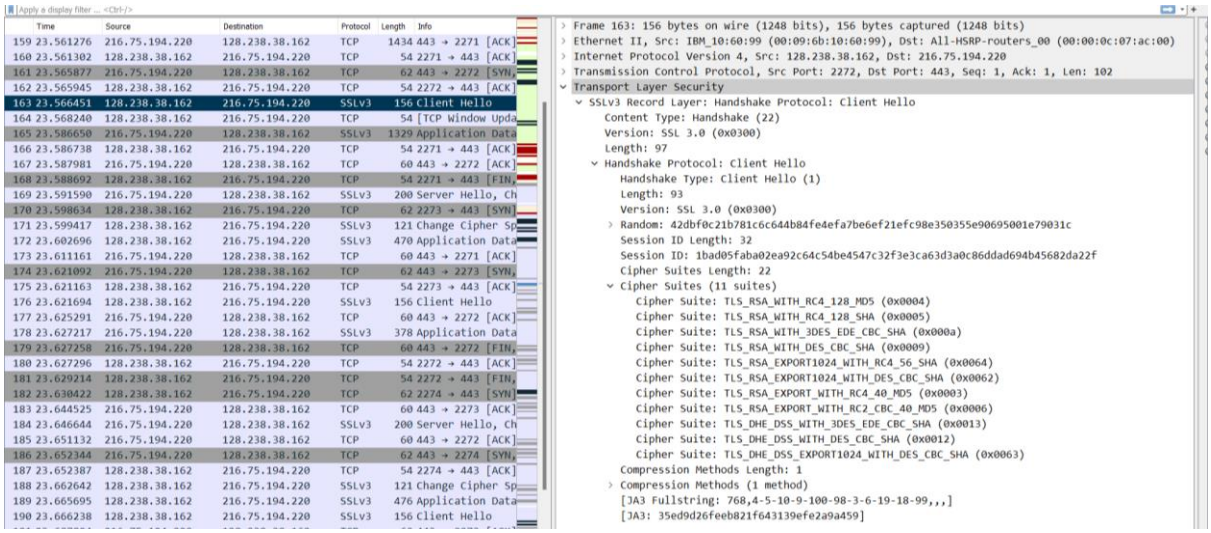
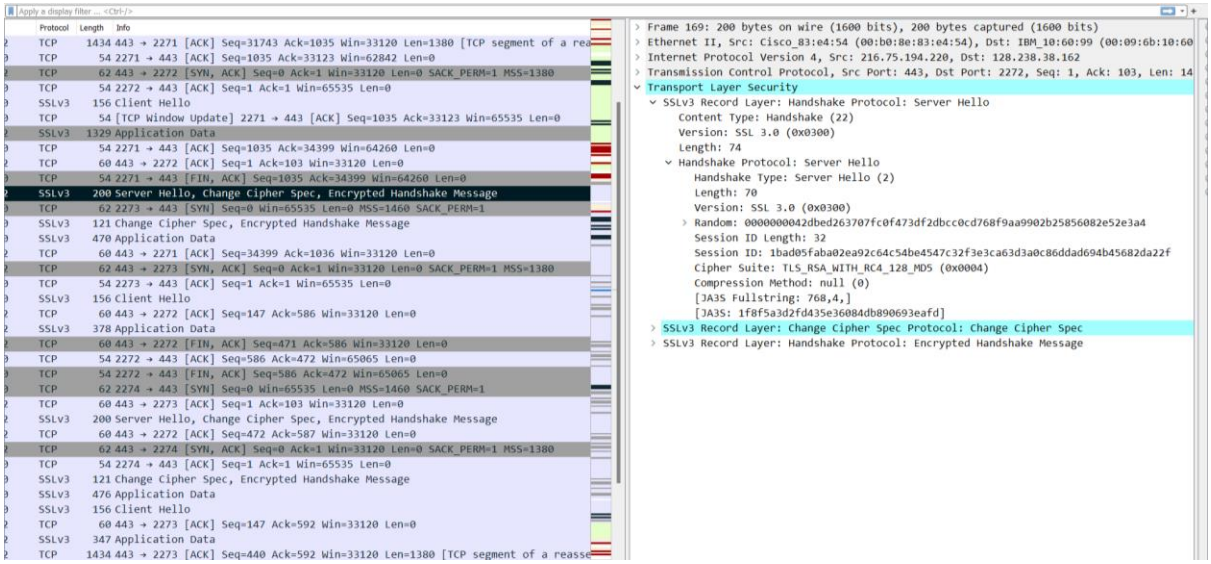


Figure 6

Snapshot of Server Hello TLS Description



CHAPTER 3

Figure 7

Snapshot of AS-REQ Description

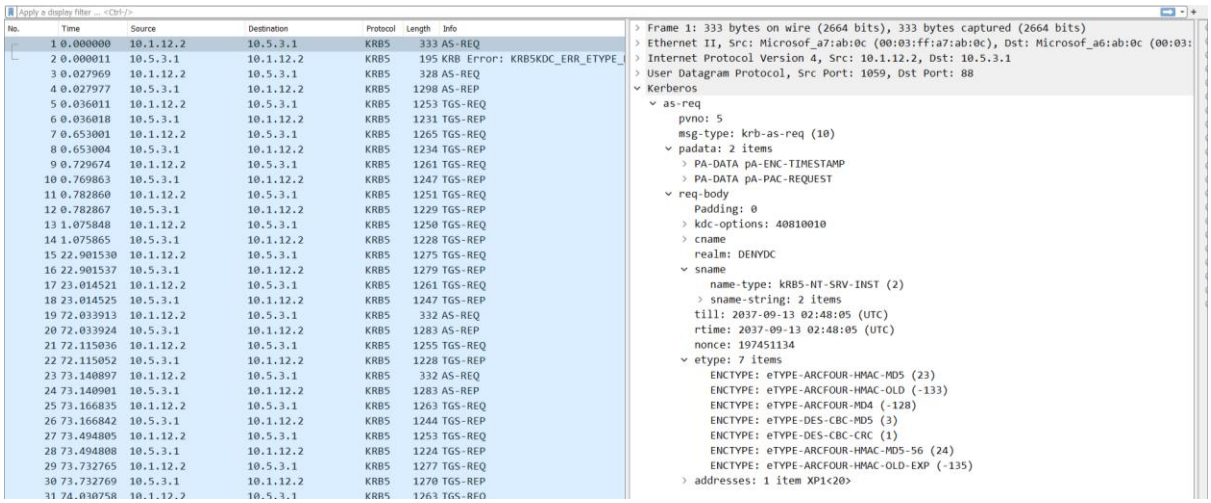


Figure 8

Snapshot of AS-REP Description

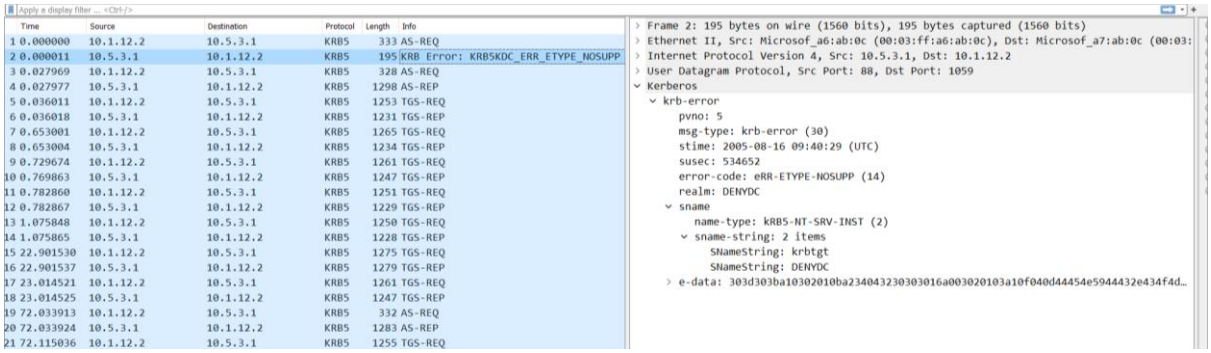


Figure 9

Snapshot of KRB-ERROR Description

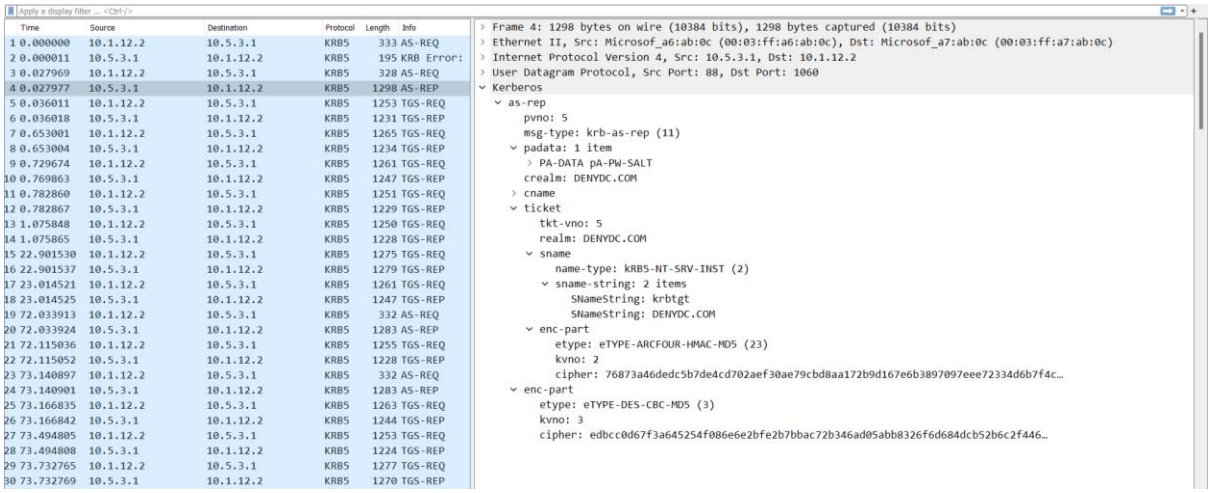


Figure 10

Snapshot of TGS-REQ Description

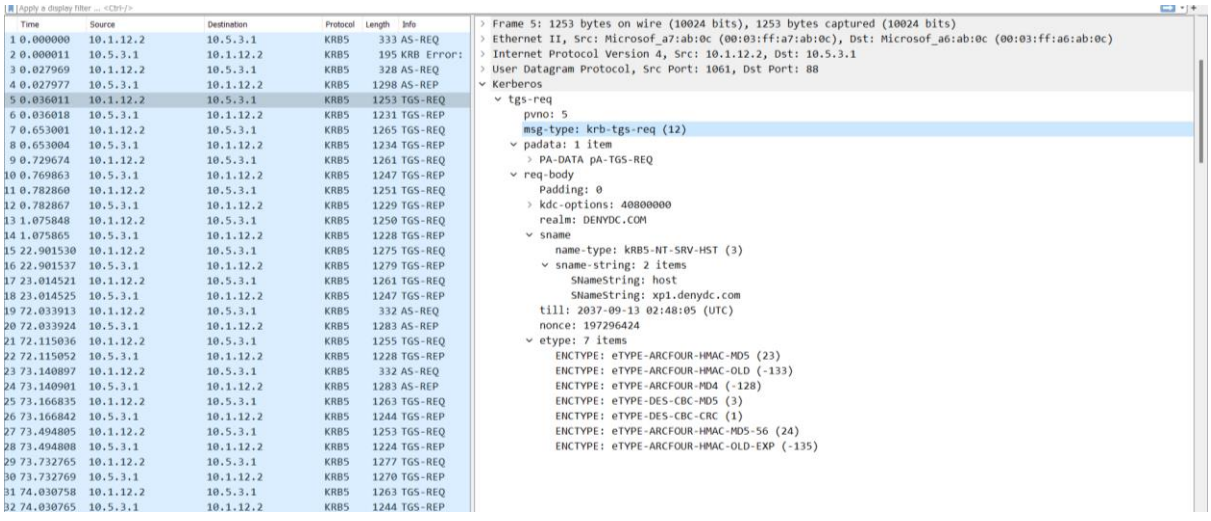
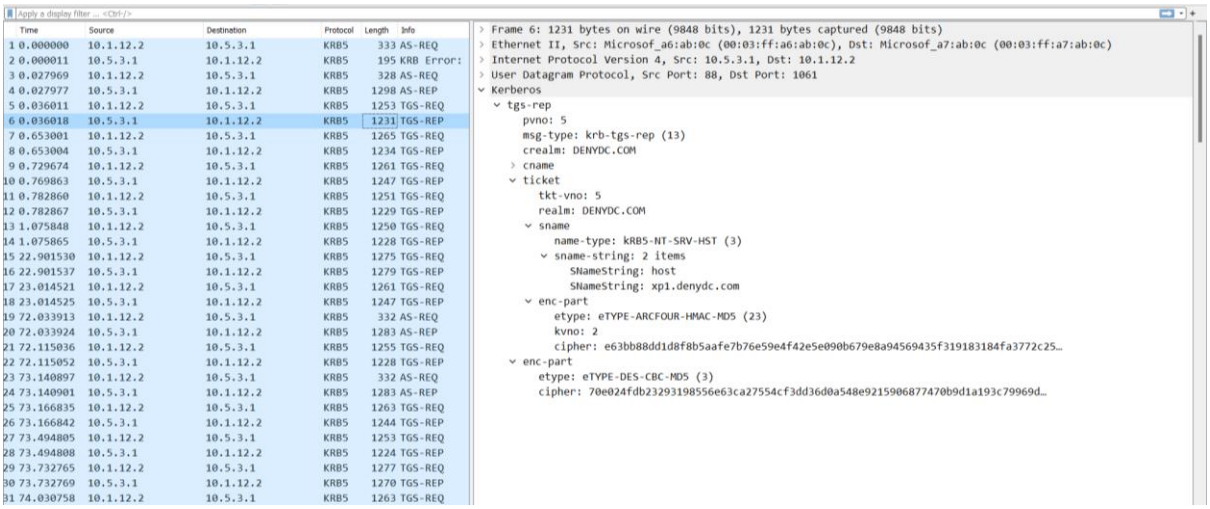


Figure 11

Snapshot of TGS-REP Description



CHAPTER 4

Figure 12

Snapshot of DNS Standard Query Description

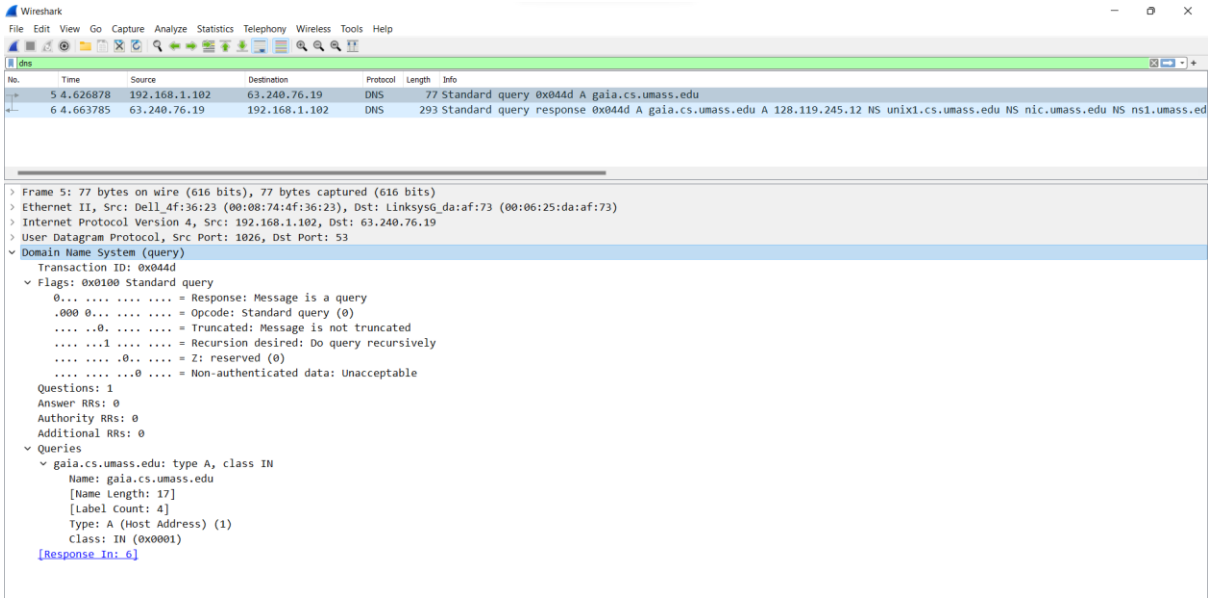


Figure 13

Snapshot of DNS Standard Query Response Description