Security Protocols Portfolio

TCP Handshake

The Transmission Control Protocol is a transport layer technology, and internet connected devices use this connection-oriented protocol standard to help communicate over it[1]. It first establishes the connection between 2 devices using the handshake, before reliably exchanging data[2]. It supports integrity of the data by having packet flow control which detects duplicate, missing and unordered packets[3].

Figure 1 shows a TCP handshake that was captured on Wireshark, it has 3 stages: SYN, SYN/ACK and ACK.

The source IP address of "192.168.1.102" belongs to the client who requested the connection, and the server's address is 128.119.245.12. The port number is used alongside the address to provide a unique endpoint of a device in the session[4]. There are ports reserved for certain protocols to use (0-1023)[5]. The first message identifies that it is being sent from the client's port 4127 (user port), to the server's port 80 which corresponds to HTTP[6].

Handshake overview

SYN:
The client picks an initial sequence number (ISN), and then sends a SYN TCP packet containing the ISN.
SYN/ACK:
The server receives this, and send back a SYN/ACK comprising of two parts: their own picked ISN, and an acknowledgement number of having received the client's ISN which is the client's ISN incremented by 1.
ACK:
The client receives the SYN/ACK and sends the acknowledgement that they successfully received the server's ISN by sending the server's ISN incremented by 1.
After this, the communication is established.
Sequence numbers keep track of the packets to ensure reliable data flow.
During the handshake, connection parameters can be negotiated for optimisation[7].

Wireshark analysis

Figure 2 shows the SYN stage. We can extract the following information from the labelled boxes:

1. Confirms the IP addresses of the Client and Server
2. Confirms the ports used by the Client and Server
3. Shows the ISN picked by the client (4113720497)
4. Highlights the flags set on the packet and confirms it is only a SYN packet.
5. Shows the connection parameters that the client supports.

Figure 3 shows the SYN/ACK stage. We can extract the following information:

1. The server picks its ISN (1806062737)
2. The server takes the client's ISN from the SYN packet, increments it (4113720498) and sends it as acknowledgment
3. Both the SYN and ACK flag are set
4. Shows the options the server can support.

Figure 4 shows the ACK stage. We can extract the following information:

1. The client takes the server's ISN, increments it (1806062738) and sends it as acknowledgement
2. It only has the ACK flag set

Security aspects

The handshake is unencrypted, so is transmitted in plaintext which voids confidentiality. It attempts at integrity with the ISNs, however if active attacks are attempted, such as MITM, then there is no protection against it[8]. In terms of availability, it has a weakness to DOS attacks where an attacker can cause a 'SYN flood' and exhaust the server's resources – but there are mitigation for this, such as SYN cookies[9].

SSL/TLS Cipher Suites

To evaluate the differences between the cipher suites of SSLv2 and SSLv3, we will analyse a Wireshark capture of SSL exchanges from 2005 (Figure 18).
Figure 5 shows a Client Hello message using SSLv2 (deprecated in 2011) and Figure 6 shows a Client Hello message using SSLv3 (deprecated in 2015)[10]. The red boxes highlights the cipher suites the respective client supports.

The SSLv3 packet supports all the same cipher suites as SSLv2, except for 6 that utilise the SSL2 protocol specifically and not TLS.

Security strengths of the cipher suite options

Protocols(TLS,SSLv2)

- As SSLv2 was deprecated with security vulnerabilities, their ciphers should not be used[11].
- Only cipher suites using TLS 1.2 or 1.3 should be chosen as earlier versions were deprecated from 2021[12][13].

Key exchange algorithms (RSA, DHE, EXPORTRSA)

- RSA is an asymmetric encryption algorithm that is no longer used from TLS1.3. It does not have the advantage of forward secrecy but it does ensure confidentiality and integrity[14][15].
- DHE does not ensure authentication, so is vulnerable to MITM, but it guarantees forward secrecy[16].
- Export RSA purposefully restricted the key sizes making them vulnerable to brute force attacks due to their short key lengths and are deprecated mostly now[17][18].

Cipher algorithms(RC4, 3DES, RC2, DES)

- RC4 is a symmetric stream cipher and has known keystream bias and design weaknesses[19][20].
- RC2 is a symmetric block cipher and uses a small key and block size making it vulnerable to brute force and birthday attacks[21][22].
- 3DES is a symmetric block cipher that applies DES 3 times to each block and was deprecated in 2023 due to having security vulnerabilities and weak key size making it vulnerable to brute force attacks [23][24]. DES was withdrawn in 2005[25].
- A better alternative than all these options is to use AES which is a block cipher that is used to secure modern technology[26].

Hash functions(MD5, SHA):

- MD5 and SHA-1 are cryptographically broken due to collision weaknesses[27][28]. A better alternative is to use SHA-256 or higher, as it is used widely in modern applications and is considered secure[29].

Figure 7 shows a Client Hello message and Figure 8 shows the corresponding Server Hello message. The server picked “TLS_RSA_WITH_RC4_128_MD5” which is RSA with 128 bit RC4 stream cipher and MD5 hash. It is not the strongest out of the list, so the server must have picked it based on its own configuration set by an admin[30]. For example, options with the use of SHA would be considered stronger[31]. However, in 2008, MD5 was officially declared broken, but in 2005, it was still supported[27].

The admin here could've prioritised performance, as RC4 is lighter and faster as it is a stream cipher, compared to block ciphers like 3DES[32]. The server could also be using an older OS that supports a limited set of cipher suites. If TLS 1.0 is being used, then RC4 may have been picked due to it being the most supported at the time[33].

Kerberos

The below describes how the Kerberos protocol works whilst referring to labels on the respective figures:

- A user wants to access a service.
- Figure 9 shows that the client proves the user's identity to the Authentication Service (AS) through an AS request. We find that: the Kerberos version is 5 (1),the pre-authentication data value is the timestamp encrypted with the user's password which prevents replay attacks(2), the client's name is "DES"(3), the domain's name is "DENDYC"(4), the client requested an expiry time of the future Ticket Granting Ticket(TGT) (5), there is a nonce to ensure the session is unique and protected against replay attacks(6), the client listed its supported encryption types(7).
- Figure 10 shows the AS response where it returns a TGT(2) that is encrypted with the Ticket Granting Server's(TGS) secret key along with RC4 and HMAC with MD5. RC4 has known keystream bias and design weaknesses[19][20]. AES is the modern secure solution so should be used[26]. MD5 has weak collision resistance, and SHA-256 should be used as it is the modern secure solution[27][29].

  The AS also returns a TGS session key(3) that is encrypted using a secret key that consists of the user's password and salt value(1), and DES in CBC mode using MD5. DES has a short key length so is vulnerable to brute force attacks[34]. CBC is better than ECB as it has better pattern concealment[35]. The client does not need to renew this TGT until a new user session or expiry.
- The client retrieves the TGS session key using its secret key, and Figure 11 shows the client's TGS request, where the client sends the encrypted TGT(1), with an authenticator(2) encrypted with the TGS session key(that includes details such as the client id and timestamp), to the TGS. The service requested is "host/xp1.denydc.com"(3) and label 4 lists the encryptions that the client supports.
- The TGS verifies the user and sends a TGS response as shown in Figure 12. The TGS returns a service ticket encrypted with the service's secret key(1), along with a service session key encrypted with the TGS' session key(2). This service ticket interaction is repeated every time the user want to access a new service or if it expires.
- The client gives the service ticket with an authenticator (encrypted with the service session key) to the Service Server (SS)[35][36]
- The SS decrypts and verifies the ticket and grants access if valid[36].

Kerberos prevents having every server store every user's password which decreases the attack surface as AS is a singular point of authentication that knows all the passwords, but it also is the single point of failure and requires high level of physical security as it becomes the core target[38][39].

This example was done over UDP rather than TCP because UDP is connectionless and has less overhead, so has better performance, and is suited for the small message exchanges[40].

The Kerberos messages are transmitted in plaintext which voids confidentiality. Integrity and authentication are supported with the session and secret key handling, and there is forward secrecy. A weakness is that Kerberos relies on the user's password and machine being secure as the initial means of authentication.
[41][42]

DNS Anomalies

Overview

Computers communicate with each other via their IP addresses. Users use domain names to easily access websites, which is the user friendly mapping of an address, such as "www.google.com". To retrieve the correct address, computers use a domain name system (DNS) which is a distributed hierarchical database considered to be the "phonebook of the internet" [43].
When a client queries a domain name, the DNS resolver checks if the address is in the computer's local DNS cache. If so, it returns the IP address. If not, it sends the query onto the recursive DNS resolver which checks its cache, and if not found, it queries the root server which refers the TLD (Top Level Domain) DNS server. The recursive resolver queries the TLD server which refers the authoritative server (organisation's own DNS servers). Upon querying the authoritative server, the correct IP address is retrieved. [44][45]

Figure 13 shows a DNS query. We learn:

1. The query goes to a local DNS resolver that is on the same network as the client (as they both have the same network bytes of 128.238) and this checks if it has the cached result or if the request needs to be forwarded.
2. The query is using UDP instead of TCP. This is because UDP is more efficient for small packets like this and is lightweight and connectionless allowing for faster querying[39].
3. The DNS assigned port is 53[46].
4. This is a query packet
5. The user is trying to access "www.ietf.org". The client expects an IPv4 address as it is type A[47].

Figure 14 shows the DNS response. We learn:

1. It is a DNS response
2. 2 results were found for the domain name.
3. There were no authority records, meaning the query was not forwarded to the authoritative server. This, along with the short time response(5) strongly suggest that the addresses were stored in the cache of the DNS resolver.
4. The found addresses were: 132.151.6.75 and 65.246.255.51. A domain may have more than one IP address due to load balancing and failover support[48][49].

DNS Anomaly

Figure 15 shows anomalies in communication on DNS port 53 and that a TCP connection is trying to be maintained on port 53 instead of port 80/443. This suggests a remote shell being used on the infected victim's machine (192.168.1.3) where the shell is trying to use a covert channel to communicate with the attacker's server (192.168.1.2)[50][51]. The reason for using port 53, is because DNS is mostly always allowed through firewalls, and they wouldn't be monitored as closely as port 80/443[52]. An anomaly here is "Dup ACK" which appears many times and is normally sent when packets are detected out of order or are lost[53][54], suggesting that the attacker is trying to maintain a stable connection, but is struggling due to network issues, or network devices not being able to handle TCP packets well when it is expecting DNS packets.

There are signs of this attack in an earlier DNS request-response pair before the connection was attempted, where the victim's machine (through remote control of the hacker) was trying to find the domain name that corresponds to the attacker's server's address. Figure 16 shows a standard DNS request for www.www.com.lan that has the suffix "lan" suggesting a local domain in a private network[55], signalling that the attacker and victim are on the same network, and this is further proven by how the victim's and attacker's addresses have the same network bytes. Figure 17 shows the response, and how there was no address answer found (answers RRs is 0). The lack of answer confirms the attacker's intention of experimenting to see if the "lan" suffix would work, as this would take advantage of generating internal traffic that is more likely to be undetected from a firewall.

References

[1] '(PDF) TCP/IP Stack Transport Layer Performance, Privacy, and Security issues', *ResearchGate*, Nov. 2024, doi: 10.30574/wjaets.2024.11.2.0098.

[2] 'Three-Way Handshake - an overview | ScienceDirect Topics'. Accessed: Mar. 18, 2025. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/three-way-handshake

[3] 'Transmission Control Protocol (TCP) (article)', Khan Academy. Accessed: Mar. 18, 2025. [Online]. Available: https://www.khanacademy.org/a/transmission-control-protocol--tcp

[4] 'Port (computer networking)', *Wikipedia*. Mar. 07, 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Port_(computer_networking)&oldid=1279220326

[5] 'List of TCP and UDP port numbers', *Wikipedia*. Mar. 17, 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=List_of_TCP_and_UDP_port_numbers&oldid=1280885199#Well-known_ports

[6] 'HTTP', *Wikipedia*. Dec. 12, 2024. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=HTTP&oldid=1262579277

[7] 'Options Field in TCP Header', GeeksforGeeks. Accessed: Mar. 18, 2025. [Online]. Available: https://www.geeksforgeeks.org/options-field-in-tcp-header/

[8] 'What Is a Man-in-the-Middle (MITM) Attack? | IBM'. Accessed: Mar. 18, 2025. [Online]. Available: https://www.ibm.com/think/topics/man-in-the-middle

[9] W. Eddy, 'TCP SYN Flooding Attacks and Common Mitigations', Internet Engineering Task Force, Request for Comments RFC 4987, Aug. 2007. doi: 10.17487/RFC4987.

[10] 'Transport Layer Security', *Wikipedia*. Mar. 15, 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Transport_Layer_Security&oldid=1280563566#History_and_development

[11] T. Polk and S. Turner, 'Prohibiting Secure Sockets Layer (SSL) Version 2.0', Internet Engineering Task Force, Request for Comments RFC 6176, Mar. 2011. doi: 10.17487/RFC6176.

[12] 'Transport Layer Security', *Wikipedia*. Mar. 15, 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Transport_Layer_Security&oldid=1280563566#History_and_development

[13] ‘Using TLS to protect data’. Accessed: Mar. 18, 2025. [Online]. Available: https://www.ncsc.gov.uk/guidance/using-tls-to-protect-data

[14] ‘Forward secrecy’, *Wikipedia*. Feb. 23, 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Forward_secrecy&oldid=1277268114

[15] ‘RSA Algorithm in Cryptography’, GeeksforGeeks. Accessed: Mar. 18, 2025. [Online]. Available: https://www.geeksforgeeks.org/rsa-algorithm-cryptography/

[16] ‘Diffie–Hellman key exchange’, *Wikipedia*. Mar. 08, 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Diffie%E2%80%93Hellman_key_exchange&oldid=1279458222

[17] ‘Export of cryptography from the United States’, *Wikipedia*. Jan. 19, 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Export_of_cryptography_from_the_United_States&oldid=1270347631

[18] C. Bartle and N. Aviram, ‘Deprecating Obsolete Key Exchange Methods in TLS 1.2’, Internet Engineering Task Force, Internet Draft draft-ietf-tls-deprecate-obsolete-kex-03, Sep. 2023. Accessed: Mar. 18, 2025. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-deprecate-obsolete-kex-03

[19] ‘RC4’, *Wikipedia*. Oct. 25, 2024. Accessed: Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=RC4&oldid=1253331780

[20] ‘RC4 Encryption Algorithm’, GeeksforGeeks. Accessed: Mar. 18, 2025. [Online]. Available: https://www.geeksforgeeks.org/rc4-encryption-algorithm/

[21] R. L. Rivest, ‘A Description of the RC2(r) Encryption Algorithm’, Internet Engineering Task Force, Request for Comments RFC 2268, Mar. 1998. doi: 10.17487/RFC2268.

[22] M. Bellare and P. Rogaway, ‘Introduction to Modern Cryptography’.

[23] I. T. L. Computer Security Division, ‘NIST to Withdraw Special Publication 800-67 Revision 2 | CSRC’, CSRC | NIST. Accessed: Mar. 19, 2025. [Online]. Available: https://csrc.nist.gov/news/2023/nist-to-withdraw-sp-800-67-rev-2

[24] ‘NVD - CVE-2016-2183’. Accessed: Mar. 19, 2025. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2016-2183

[25] ‘NIST Withdraws Outdated Data Encryption Standard’, *NIST*, Jun. 2005, Accessed: Mar. 19, 2025. [Online]. Available: https://www.nist.gov/news-events/news/2005/06/nist-withdraws-outdated-data-encryption-standard

[26] ‘(PDF) A comparison of the 3DES and AES encryption standards’, *ResearchGate*, Oct. 2024, doi: 10.14257/ijsia.2015.9.7.21.

[27] ‘CERT/CC Vulnerability Note VU#836068’. Accessed: Mar. 19, 2025. [Online]. Available: https://www.kb.cert.org

[28] G. Leurent and T. Peyrin, ‘SHA-1 is a Shambles - First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust’, 2020, 2020/014. Accessed: Mar. 19, 2025. [Online]. Available: https://eprint.iacr.org/2020/014

[29] M. Bedford Taylor, ‘The Evolution of Bitcoin Hardware’, *Computer*, vol. 50, no. 9, pp. 58–66, 2017, doi: 10.1109/MC.2017.3571056.

[30] andreipo, ‘Manage Transport Layer Security (TLS) in Windows’. Accessed: Mar. 19, 2025. [Online]. Available: https://learn.microsoft.com/en-us/windows-server/security/tls/manage-tls

[31] ‘Difference between MD5 and SHA1’, GeeksforGeeks. Accessed: Mar. 29, 2025. [Online]. Available: https://www.geeksforgeeks.org/difference-between-md5-and-sha1/

[32] S. O. Sharif and S. P. Mansoor, ‘Performance analysis of stream and block cipher algorithms’, in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, Aug. 2010, pp. V1-522-V1-525. doi: 10.1109/ICACTE.2010.5578961.

[33] H. K. Lee, T. Malkin, and E. Nahum, ‘Cryptographic strength of ssl/tls servers: current and recent practices’, in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, San Diego California USA: ACM, Oct. 2007, pp. 83–92. doi: 10.1145/1298306.1298318.

[34] ‘Data-Encryption-Standart-DES-Encyclopedia-article.pdf’. Accessed: Mar. 27, 2025. [Online]. Available: https://orbilu.uni.lu/bitstream/10993/17076/1/Data-Encryption-Standart-DES-Encyclopedia-article.pdf

[35] ‘ECB Mode vs CBC Mode in Cryptography’, GeeksforGeeks. Accessed: Mar. 27, 2025. [Online]. Available: https://www.geeksforgeeks.org/ecb-mode-vs-cbc-mode-in-cryptography/

[36] S. H. Qatinah and I. A. Al-Baltah, ‘Kerberos Protocol: Security Attacks and Solution’, in *2024 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI)*, Nov. 2024, pp. 1–7. doi: 10.1109/ICETI63946.2024.10777133.

[37] Anonymous, ‘The Kerberos Protocol Explained | Identity & Access Management’. Accessed: Mar. 27, 2025. [Online]. Available: https://iam.uconn.edu/the-kerberos-protocol-explained/

[38] ‘How Does Kerberos Work? The Authentication Protocol Explained’, freeCodeCamp.org. Accessed: Mar. 27, 2025. [Online]. Available: https://www.freecodecamp.org/news/how-does-kerberos-work-authentication-protocol/

[39] ‘What Is Kerberos? Kerberos Authentication Explained’, Fortinet. Accessed: Mar. 27, 2025. [Online]. Available: https://www.fortinet.com/uk/resources/cyberglossary/kerberos-authentication.html

[40] ‘Differences between TCP and UDP’, GeeksforGeeks. Accessed: Mar. 27, 2025. [Online]. Available: https://www.geeksforgeeks.org/differences-between-tcp-and-udp/

[41] R. Broeckelmann, ‘Kerberos Wireshark Captures: A Windows Login Example’, Medium. Accessed: Mar. 29, 2025. [Online]. Available: https://medium.com/@robert.broeckelmann/kerberos-wireshark-captures-a-windows-login-example-151fabf3375a

[42] ‘Kerberos (protocol)’, *Wikipedia*. Feb. 08, 2025. Accessed: Mar. 29, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Kerberos_(protocol)&oldid=1274612292

[43] T. Callahan, M. Allman, and M. Rabinovich, ‘On modern DNS behavior and properties’, *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 7–15, Jul. 2013, doi: 10.1145/2500098.2500100.

[44] ‘What is DNS? – Introduction to DNS - AWS’, Amazon Web Services, Inc. Accessed: Mar. 29, 2025. [Online]. Available: https://aws.amazon.com/route53/what-is-dns/

[45] ‘Working of Domain Name System (DNS) Server’, GeeksforGeeks. Accessed: Mar. 29, 2025. [Online]. Available: https://www.geeksforgeeks.org/working-of-domain-name-system-dns-server/

[46] ‘Domain names - implementation and specification’, Internet Engineering Task Force, Request for Comments RFC 1035, Nov. 1987. doi: 10.17487/RFC1035.

[47] ‘What is a DNS A record?’ Accessed: Mar. 29, 2025. [Online]. Available: https://www.cloudflare.com/learning/dns/dns-records/dns-a-record/

[48] ‘What is DNS-based load balancing? | DNS load balancing’. Accessed: Mar. 29, 2025. [Online]. Available: https://www.cloudflare.com/learning/performance/what-is-dns-load-balancing/

[49] ‘Can Single Domain Have Multiple IP Addresses?’, Uptimia.com. Accessed: Mar. 29, 2025. [Online]. Available: https://www.uptimia.com/questions/can-single-domain-have-multiple-ip-addresses

[50] ‘Detection and prevention of DNS anomalies | Infosec’. Accessed: Mar. 29, 2025. [Online]. Available: https://www.infosecinstitute.com/resources/malware-analysis/detection-prevention-dns-anomalies/

[51] techslang, ‘What is a Reverse Shell? — Definition by Techslang’, Techslang — Tech Explained in Simple Terms. Accessed: Mar. 29, 2025. [Online]. Available: https://www.techslang.com/definition/what-is-a-reverse-shell/

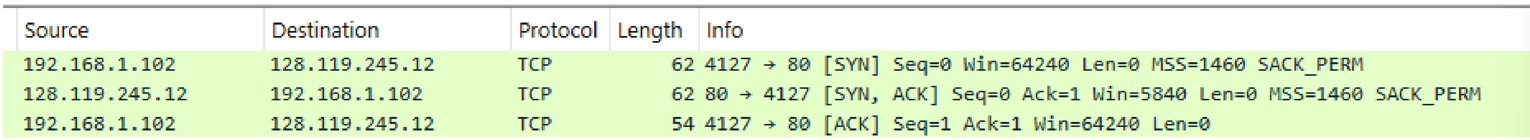
[52] ‘What is DNS Security?’, Check Point Software. Accessed: Mar. 29, 2025. [Online]. Available: https://www.checkpoint.com/cyber-hub/network-security/what-is-dns-security/

[53] ‘What Does TCP DUP ACK Mean? | Baeldung on Computer Science’. Accessed: Mar. 29, 2025. [Online]. Available: https://www.baeldung.com/cs/tcp-duplicate-acknowledgment-packet

[54] ‘Wireshark and the most common TCP issues – latebits.com’. Accessed: Mar. 29, 2025. [Online]. Available: https://latebits.com/2019/12/06/wireshark-and-the-most-common-tcp-issues/

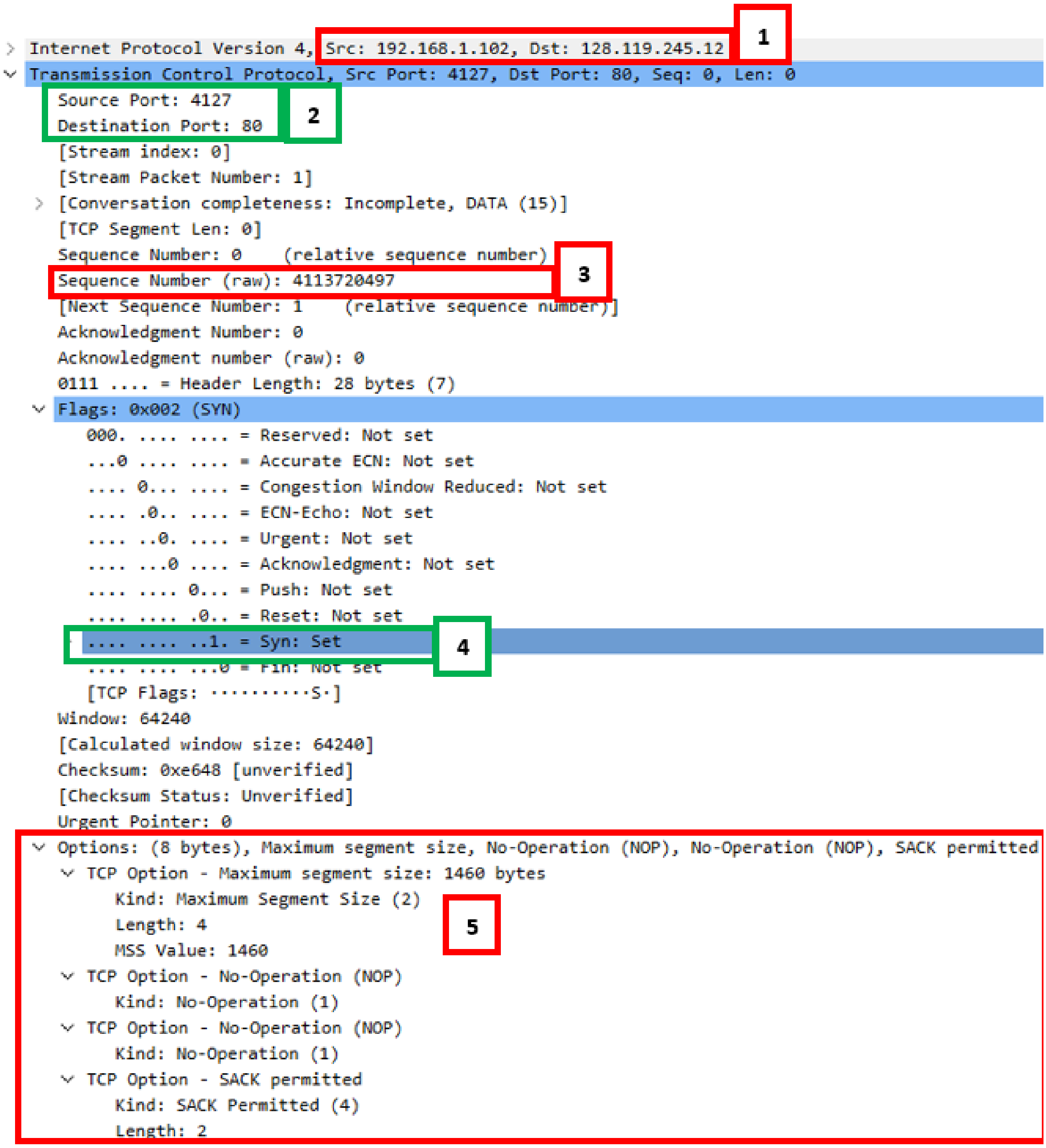
[55] maxschlepzig, ‘Answer to “What’s the difference between .local, .home, and .lan?”’, Unix & Linux Stack Exchange. Accessed: Mar. 29, 2025. [Online]. Available: https://unix.stackexchange.com/a/92517

Appendix



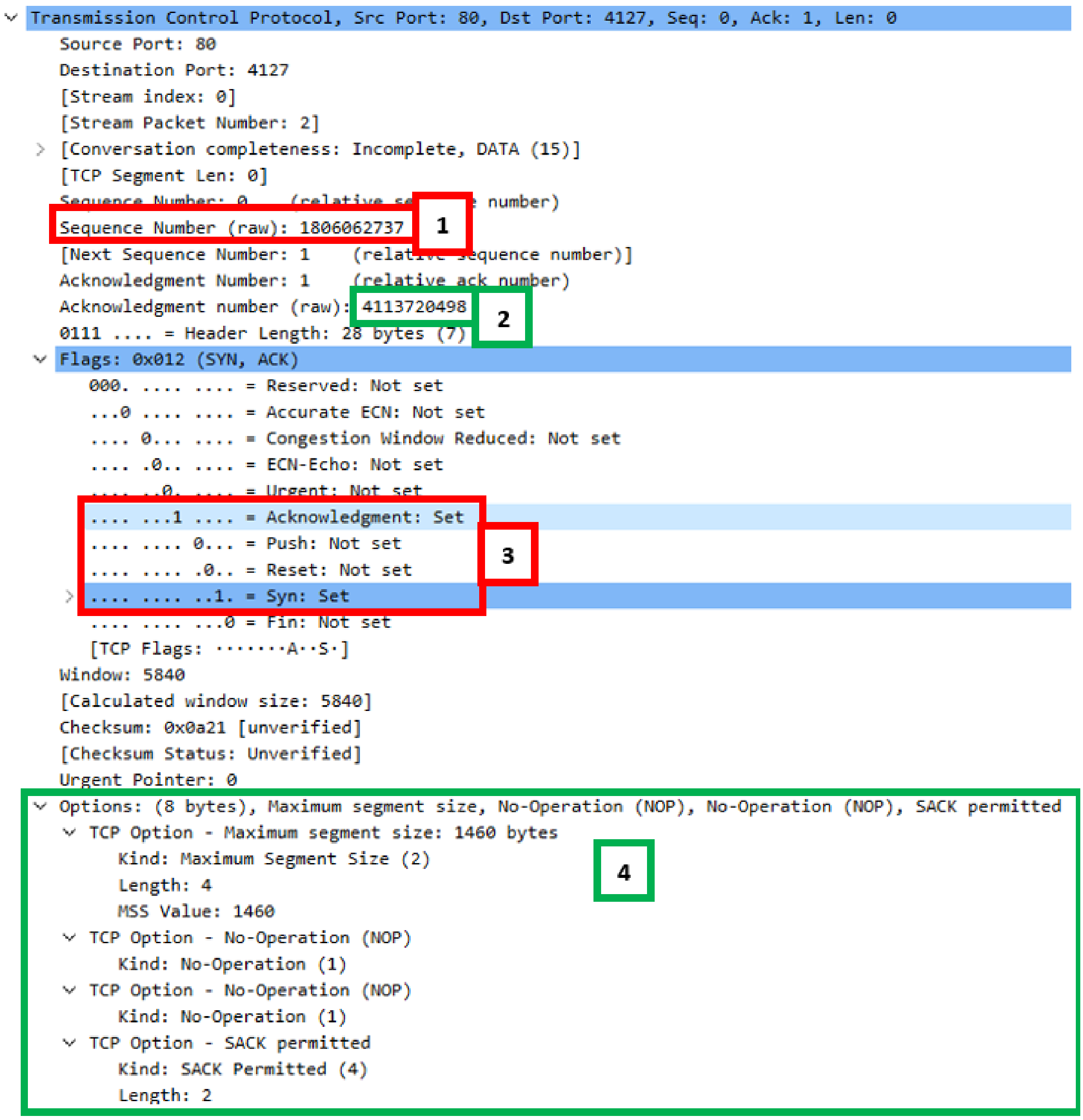
```
Source          Destination     Protocol Length Info
192.168.1.102   128.119.245.12  TCP         62 4127 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
128.119.245.12  192.168.1.102   TCP         62 80 → 4127 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
192.168.1.102   128.119.245.12  TCP         54 4127 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
```

Figure 1



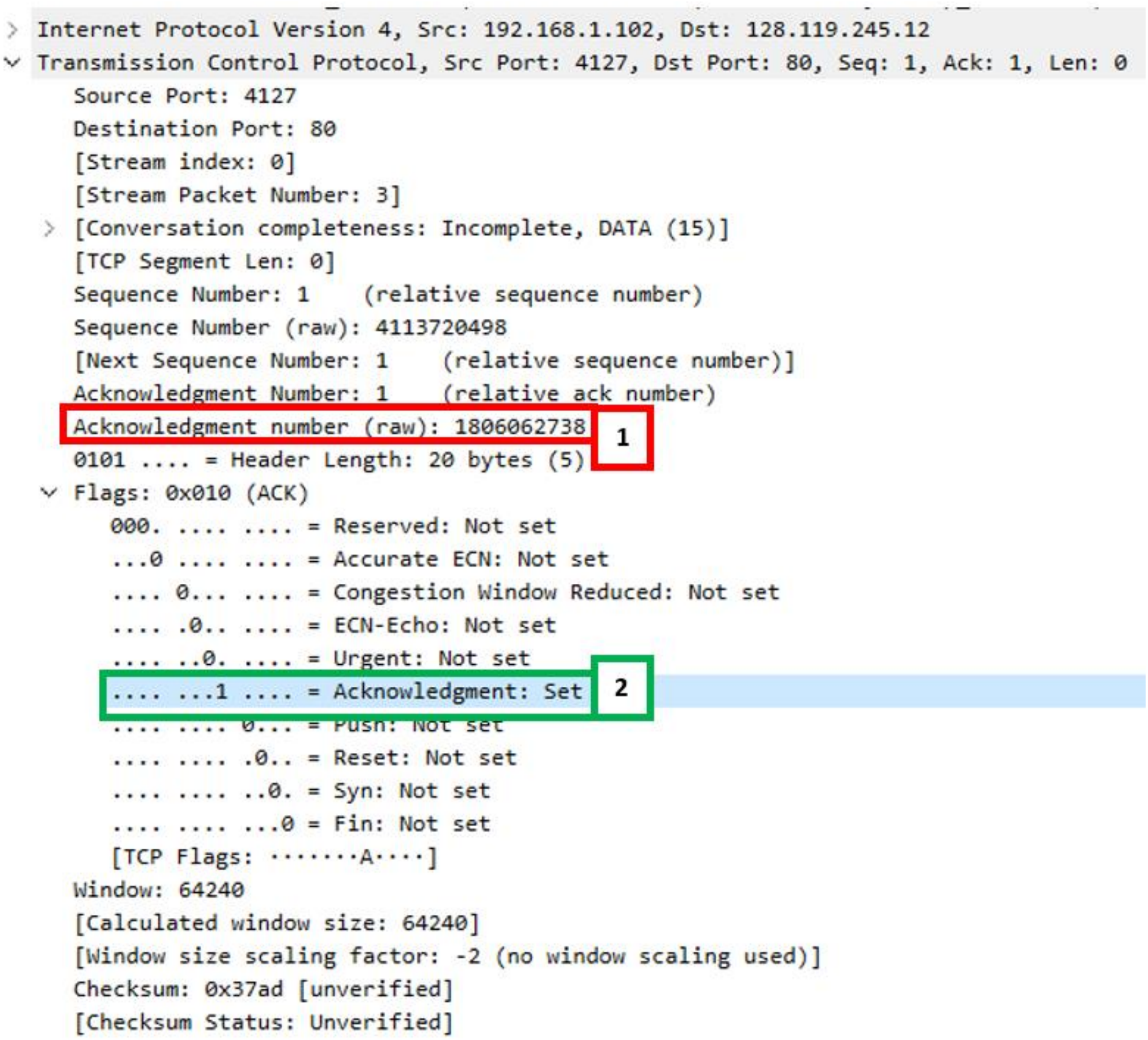
```
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 4127, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 4127
    Destination Port: 80
    [Stream index: 0]
    [Stream Packet Number: 1]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 4113720497
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    0111 .... = Header Length: 28 bytes (7)
  v Flags: 0x002 (SYN)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Accurate ECN: Not set
        .... 0... .... = Congestion Window Reduced: Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...0 .... = Acknowledgment: Not set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..1. = Syn: Set
        .... .... ...0 = Fin: Not set
        [TCP Flags: ··········S·]
    Window: 64240
    [Calculated window size: 64240]
    Checksum: 0xe648 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  v Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted
      v TCP Option - Maximum segment size: 1460 bytes
            Kind: Maximum Segment Size (2)
            Length: 4
            MSS Value: 1460
      v TCP Option - No-Operation (NOP)
            Kind: No-Operation (1)
      v TCP Option - No-Operation (NOP)
            Kind: No-Operation (1)
      v TCP Option - SACK permitted
            Kind: SACK Permitted (4)
            Length: 2
```

Figure 2



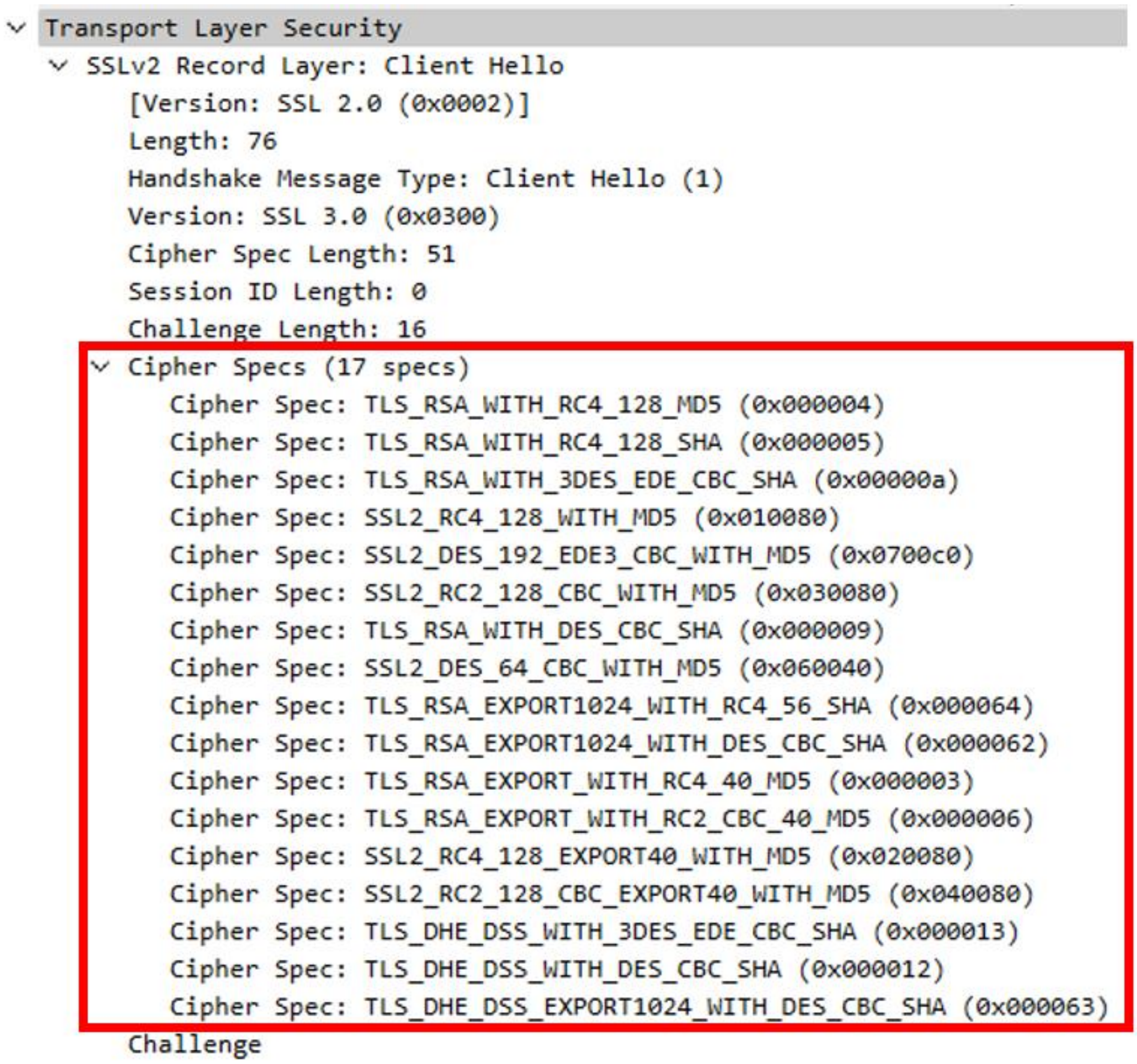
```
Transmission Control Protocol, Src Port: 80, Dst Port: 4127, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 4127
    [Stream index: 0]
    [Stream Packet Number: 2]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative se        number)
    Sequence Number (raw): 1806062737
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 4113720498
    0111 .... = Header Length: 28 bytes (7)
  v Flags: 0x012 (SYN, ACK)
       000. .... .... = Reserved: Not set
       ...0 .... .... = Accurate ECN: Not set
       .... 0... .... = Congestion Window Reduced: Not set
       .... .0.. .... = ECN-Echo: Not set
       .... ..0. .... = Urgent: Not set
       .... ...1 .... = Acknowledgment: Set
       .... .... 0... = Push: Not set
       .... .... .0.. = Reset: Not set
     > .... .... ..1. = Syn: Set
       .... .... ...0 = Fin: Not set
       [TCP Flags: ·······A··S·]
    Window: 5840
    [Calculated window size: 5840]
    Checksum: 0x0a21 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  v Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted
     v TCP Option - Maximum segment size: 1460 bytes
          Kind: Maximum Segment Size (2)
          Length: 4
          MSS Value: 1460
     v TCP Option - No-Operation (NOP)
          Kind: No-Operation (1)
     v TCP Option - No-Operation (NOP)
          Kind: No-Operation (1)
     v TCP Option - SACK permitted
          Kind: SACK Permitted (4)
          Length: 2
```

Figure 3



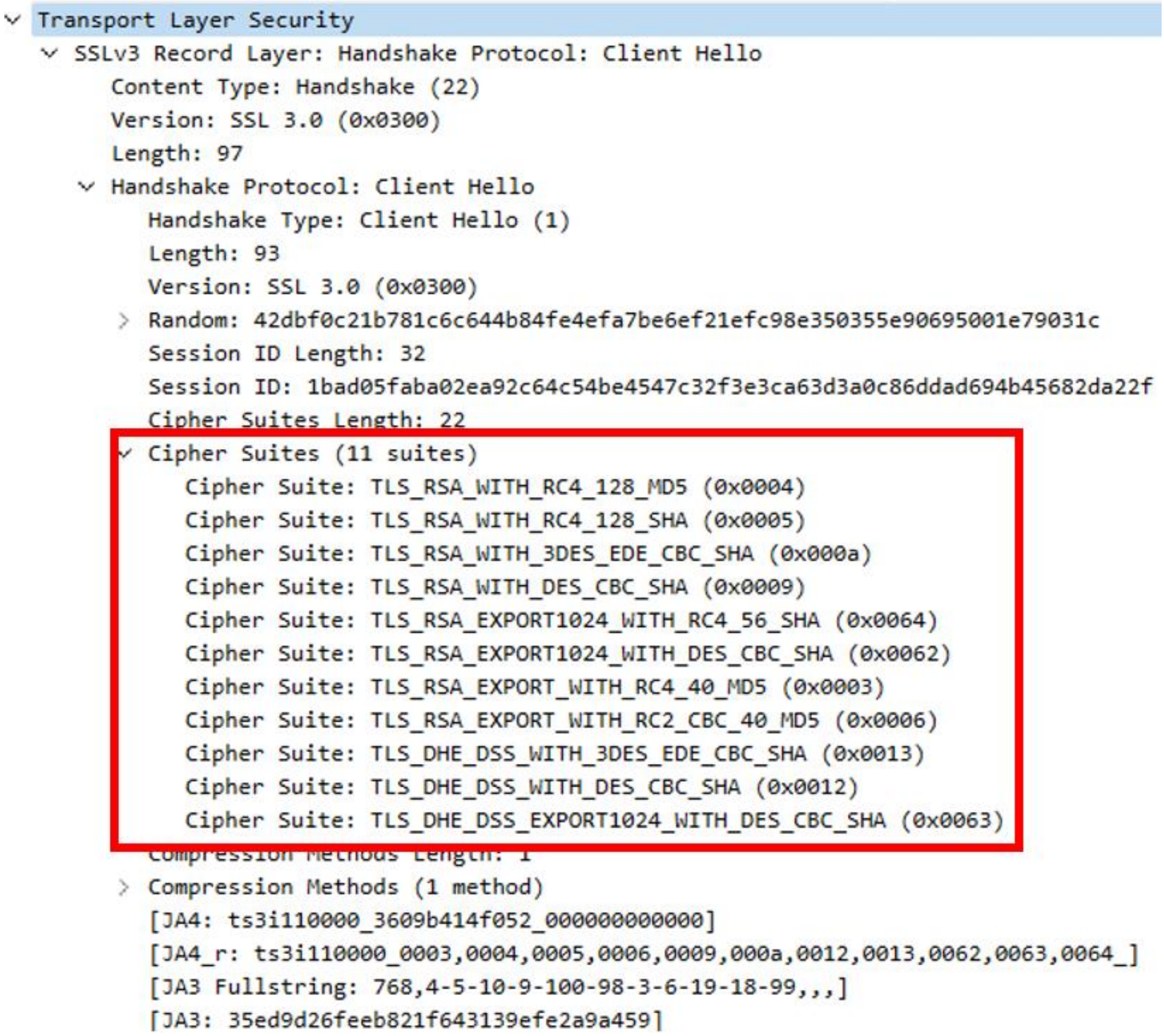
```
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 4127, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 4127
    Destination Port: 80
    [Stream index: 0]
    [Stream Packet Number: 3]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 4113720498
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1806062738
    0101 .... = Header Length: 20 bytes (5)
  v Flags: 0x010 (ACK)
       000. .... .... = Reserved: Not set
       ...0 .... .... = Accurate ECN: Not set
       .... 0... .... = Congestion Window Reduced: Not set
       .... .0.. .... = ECN-Echo: Not set
       .... ..0. .... = Urgent: Not set
       .... ...1 .... = Acknowledgment: Set
       .... .... 0... = Push: Not set
       .... .... .0.. = Reset: Not set
       .... .... ..0. = Syn: Not set
       .... .... ...0 = Fin: Not set
       [TCP Flags: ·······A····]
    Window: 64240
    [Calculated window size: 64240]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x37ad [unverified]
    [Checksum Status: Unverified]
```

Figure 4



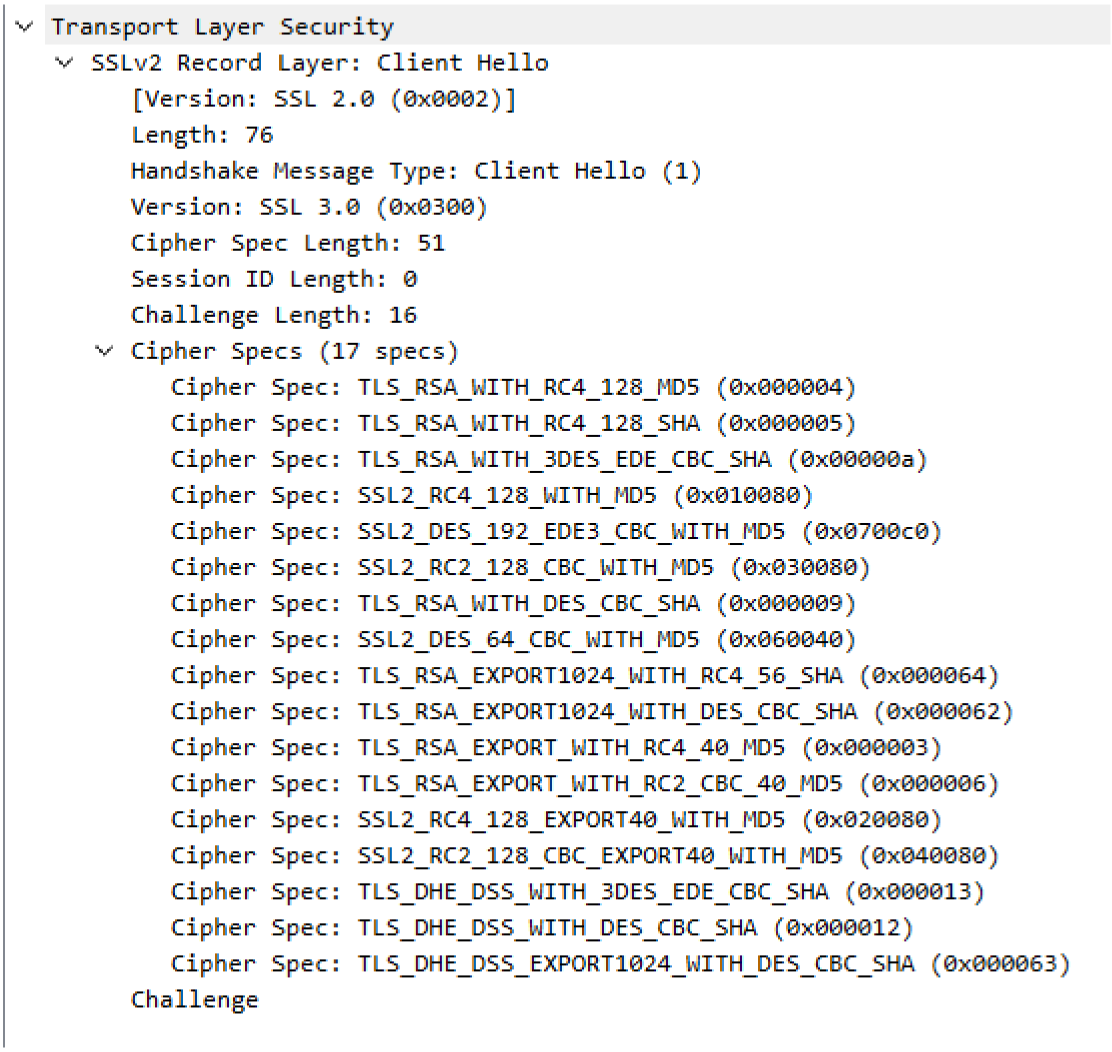
```
v Transport Layer Security
  v SSLv2 Record Layer: Client Hello
      [Version: SSL 2.0 (0x0002)]
      Length: 76
      Handshake Message Type: Client Hello (1)
      Version: SSL 3.0 (0x0300)
      Cipher Spec Length: 51
      Session ID Length: 0
      Challenge Length: 16
    v Cipher Specs (17 specs)
        Cipher Spec: TLS_RSA_WITH_RC4_128_MD5 (0x000004)
        Cipher Spec: TLS_RSA_WITH_RC4_128_SHA (0x000005)
        Cipher Spec: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x00000a)
        Cipher Spec: SSL2_RC4_128_WITH_MD5 (0x010080)
        Cipher Spec: SSL2_DES_192_EDE3_CBC_WITH_MD5 (0x0700c0)
        Cipher Spec: SSL2_RC2_128_CBC_WITH_MD5 (0x030080)
        Cipher Spec: TLS_RSA_WITH_DES_CBC_SHA (0x000009)
        Cipher Spec: SSL2_DES_64_CBC_WITH_MD5 (0x060040)
        Cipher Spec: TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x000064)
        Cipher Spec: TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x000062)
        Cipher Spec: TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x000003)
        Cipher Spec: TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (0x000006)
        Cipher Spec: SSL2_RC4_128_EXPORT40_WITH_MD5 (0x020080)
        Cipher Spec: SSL2_RC2_128_CBC_EXPORT40_WITH_MD5 (0x040080)
        Cipher Spec: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x000013)
        Cipher Spec: TLS_DHE_DSS_WITH_DES_CBC_SHA (0x000012)
        Cipher Spec: TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA (0x000063)
      Challenge
```

Figure 5



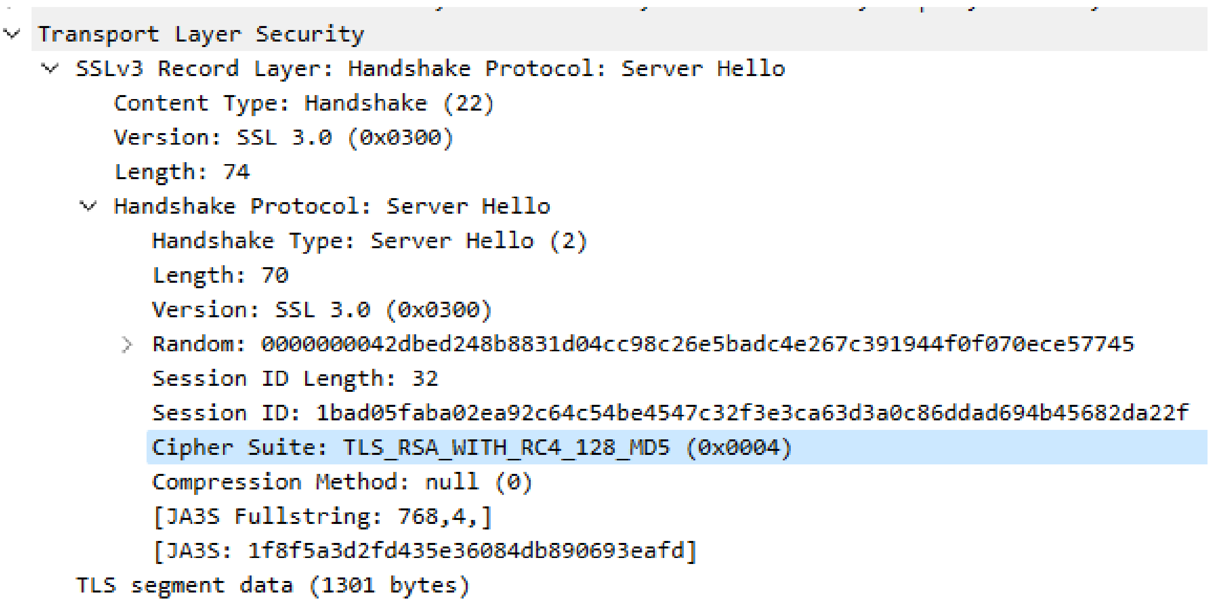
```
v Transport Layer Security
  v SSLv3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 97
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 93
        Version: SSL 3.0 (0x0300)
      > Random: 42dbf0c21b781c6c644b84fe4efa7be6ef21efc98e350355e90695001e79031c
        Session ID Length: 32
        Session ID: 1bad05faba02ea92c64c54be4547c32f3e3ca63d3a0c86ddad694b45682da22f
        Cipher Suites Length: 22
      v Cipher Suites (11 suites)
            Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
            Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
            Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
            Cipher Suite: TLS_RSA_WITH_DES_CBC_SHA (0x0009)
            Cipher Suite: TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x0064)
            Cipher Suite: TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x0062)
            Cipher Suite: TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x0003)
            Cipher Suite: TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (0x0006)
            Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
            Cipher Suite: TLS_DHE_DSS_WITH_DES_CBC_SHA (0x0012)
            Cipher Suite: TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA (0x0063)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        [JA4: ts3i110000_3609b414f052_000000000000]
        [JA4_r: ts3i110000_0003,0004,0005,0006,0009,000a,0012,0013,0062,0063,0064_]
        [JA3 Fullstring: 768,4-5-10-9-100-98-3-6-19-18-99,,,]
        [JA3: 35ed9d26feeb821f643139efe2a9a459]
```

Figure 6



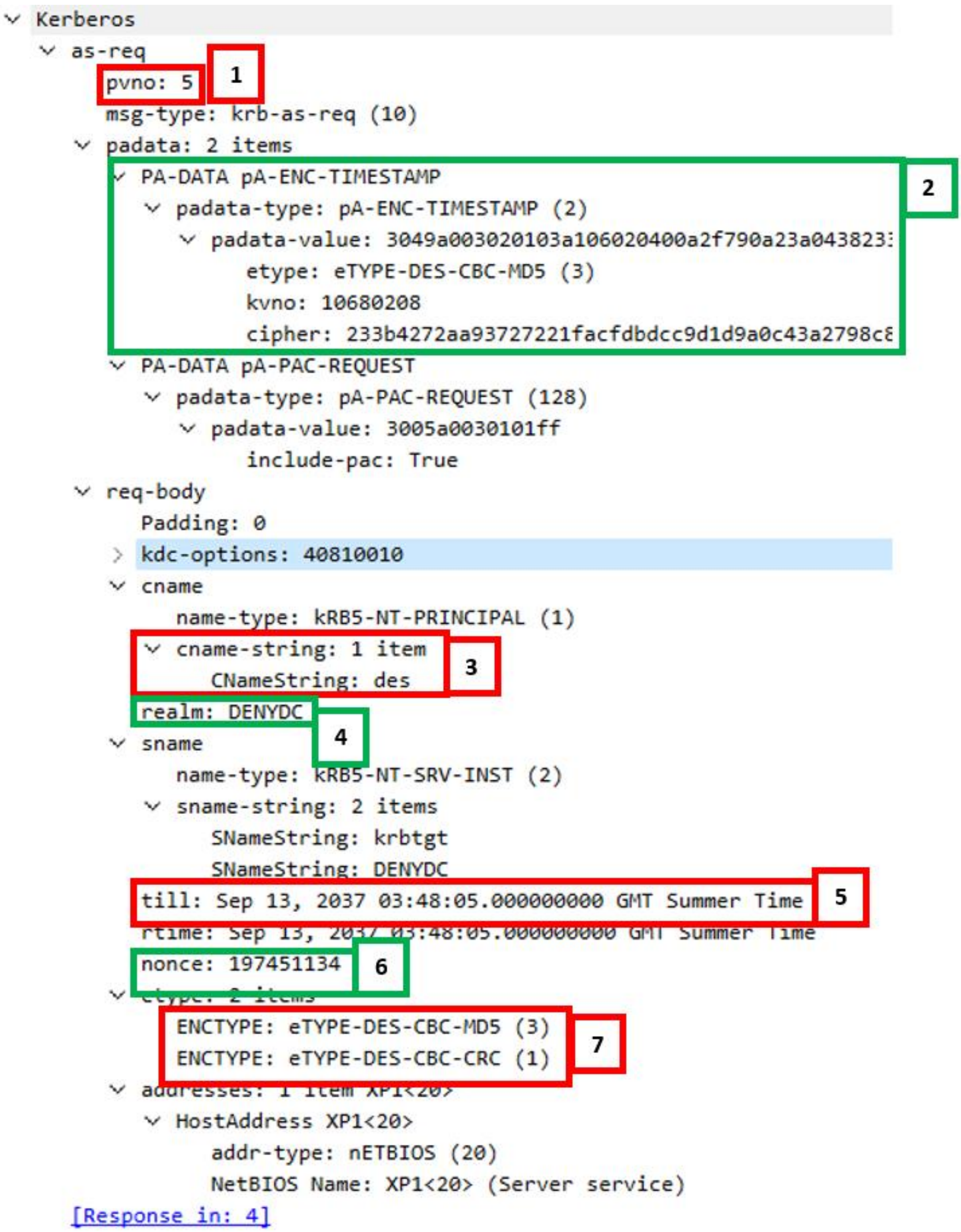
```
Transport Layer Security
  SSLv2 Record Layer: Client Hello
      [Version: SSL 2.0 (0x0002)]
      Length: 76
      Handshake Message Type: Client Hello (1)
      Version: SSL 3.0 (0x0300)
      Cipher Spec Length: 51
      Session ID Length: 0
      Challenge Length: 16
    Cipher Specs (17 specs)
        Cipher Spec: TLS_RSA_WITH_RC4_128_MD5 (0x000004)
        Cipher Spec: TLS_RSA_WITH_RC4_128_SHA (0x000005)
        Cipher Spec: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x00000a)
        Cipher Spec: SSL2_RC4_128_WITH_MD5 (0x010080)
        Cipher Spec: SSL2_DES_192_EDE3_CBC_WITH_MD5 (0x0700c0)
        Cipher Spec: SSL2_RC2_128_CBC_WITH_MD5 (0x030080)
        Cipher Spec: TLS_RSA_WITH_DES_CBC_SHA (0x000009)
        Cipher Spec: SSL2_DES_64_CBC_WITH_MD5 (0x060040)
        Cipher Spec: TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x000064)
        Cipher Spec: TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x000062)
        Cipher Spec: TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x000003)
        Cipher Spec: TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (0x000006)
        Cipher Spec: SSL2_RC4_128_EXPORT40_WITH_MD5 (0x020080)
        Cipher Spec: SSL2_RC2_128_CBC_EXPORT40_WITH_MD5 (0x040080)
        Cipher Spec: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x000013)
        Cipher Spec: TLS_DHE_DSS_WITH_DES_CBC_SHA (0x000012)
        Cipher Spec: TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA (0x000063)
      Challenge
```

Figure 7



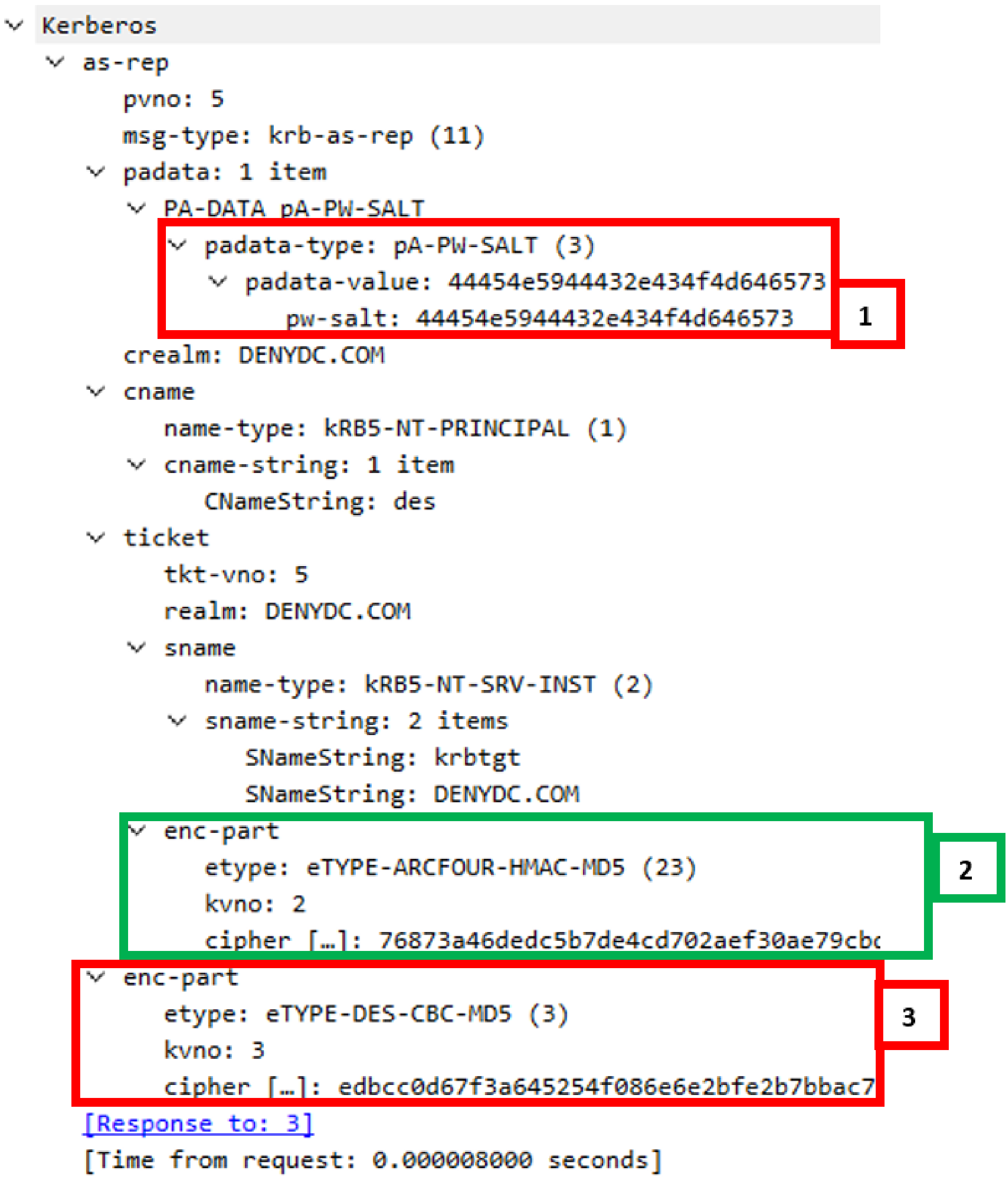
```
Transport Layer Security
  SSLv3 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 74
    Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 70
        Version: SSL 3.0 (0x0300)
        Random: 0000000042dbed248b8831d04cc98c26e5badc4e267c391944f0f070ece57745
        Session ID Length: 32
        Session ID: 1bad05faba02ea92c64c54be4547c32f3e3ca63d3a0c86ddad694b45682da22f
        Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
        Compression Method: null (0)
        [JA3S Fullstring: 768,4,]
        [JA3S: 1f8f5a3d2fd435e36084db890693eafd]
  TLS segment data (1301 bytes)
```

Figure 8



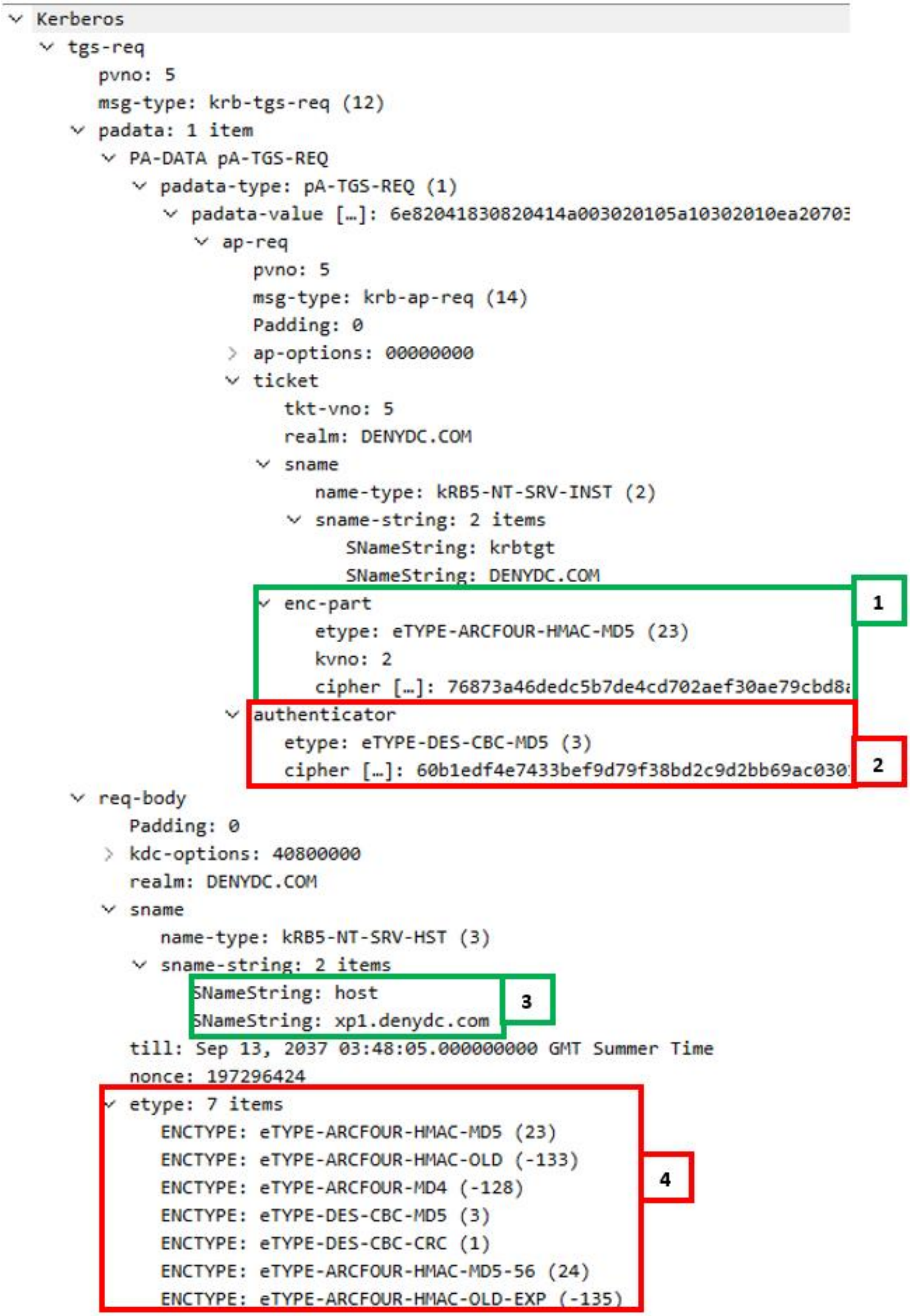
```
∨ Kerberos
  ∨ as-req
      pvno: 5
      msg-type: krb-as-req (10)
    ∨ padata: 2 items
      ∨ PA-DATA pA-ENC-TIMESTAMP
        ∨ padata-type: pA-ENC-TIMESTAMP (2)
          ∨ padata-value: 3049a003020103a106020400a2f790a23a043823
              etype: eTYPE-DES-CBC-MD5 (3)
              kvno: 10680208
              cipher: 233b4272aa93727221facfdbdcc9d1d9a0c43a2798c8
      ∨ PA-DATA pA-PAC-REQUEST
        ∨ padata-type: pA-PAC-REQUEST (128)
          ∨ padata-value: 3005a0030101ff
              include-pac: True
    ∨ req-body
        Padding: 0
      > kdc-options: 40810010
      ∨ cname
          name-type: kRB5-NT-PRINCIPAL (1)
        ∨ cname-string: 1 item
            CNameString: des
        realm: DENYDC
      ∨ sname
          name-type: kRB5-NT-SRV-INST (2)
        ∨ sname-string: 2 items
            SNameString: krbtgt
            SNameString: DENYDC
        till: Sep 13, 2037 03:48:05.000000000 GMT Summer Time
        rtime: Sep 13, 2037 03:48:05.000000000 GMT Summer Time
        nonce: 197451134
      ∨ etype: 2 items
          ENCTYPE: eTYPE-DES-CBC-MD5 (3)
          ENCTYPE: eTYPE-DES-CBC-CRC (1)
      ∨ addresses: 1 item XP1<20>
        ∨ HostAddress XP1<20>
            addr-type: nETBIOS (20)
            NetBIOS Name: XP1<20> (Server service)
  [Response in: 4]
```

Figure 9



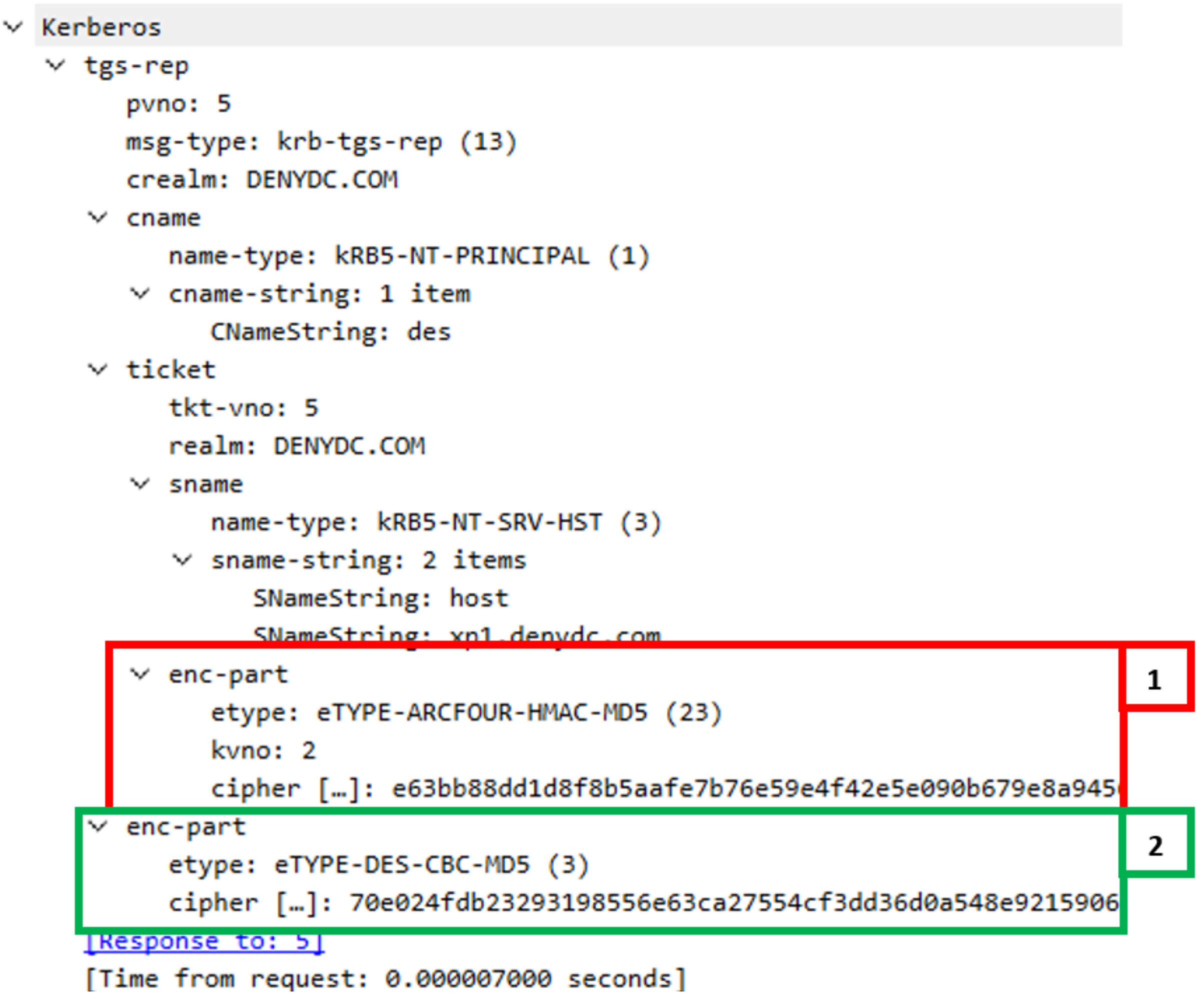
```
Kerberos
  as-rep
    pvno: 5
    msg-type: krb-as-rep (11)
    padata: 1 item
      PA-DATA pA-PW-SALT
        padata-type: pA-PW-SALT (3)
          padata-value: 44454e5944432e434f4d646573
            pw-salt: 44454e5944432e434f4d646573
    crealm: DENYDC.COM
    cname
      name-type: kRB5-NT-PRINCIPAL (1)
      cname-string: 1 item
        CNameString: des
    ticket
      tkt-vno: 5
      realm: DENYDC.COM
      sname
        name-type: kRB5-NT-SRV-INST (2)
        sname-string: 2 items
          SNameString: krbtgt
          SNameString: DENYDC.COM
      enc-part
        etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
        kvno: 2
        cipher […]: 76873a46dedc5b7de4cd702aef30ae79cbc
    enc-part
      etype: eTYPE-DES-CBC-MD5 (3)
      kvno: 3
      cipher […]: edbcc0d67f3a645254f086e6e2bfe2b7bbac7
    [Response to: 3]
    [Time from request: 0.000008000 seconds]
```

Figure 10



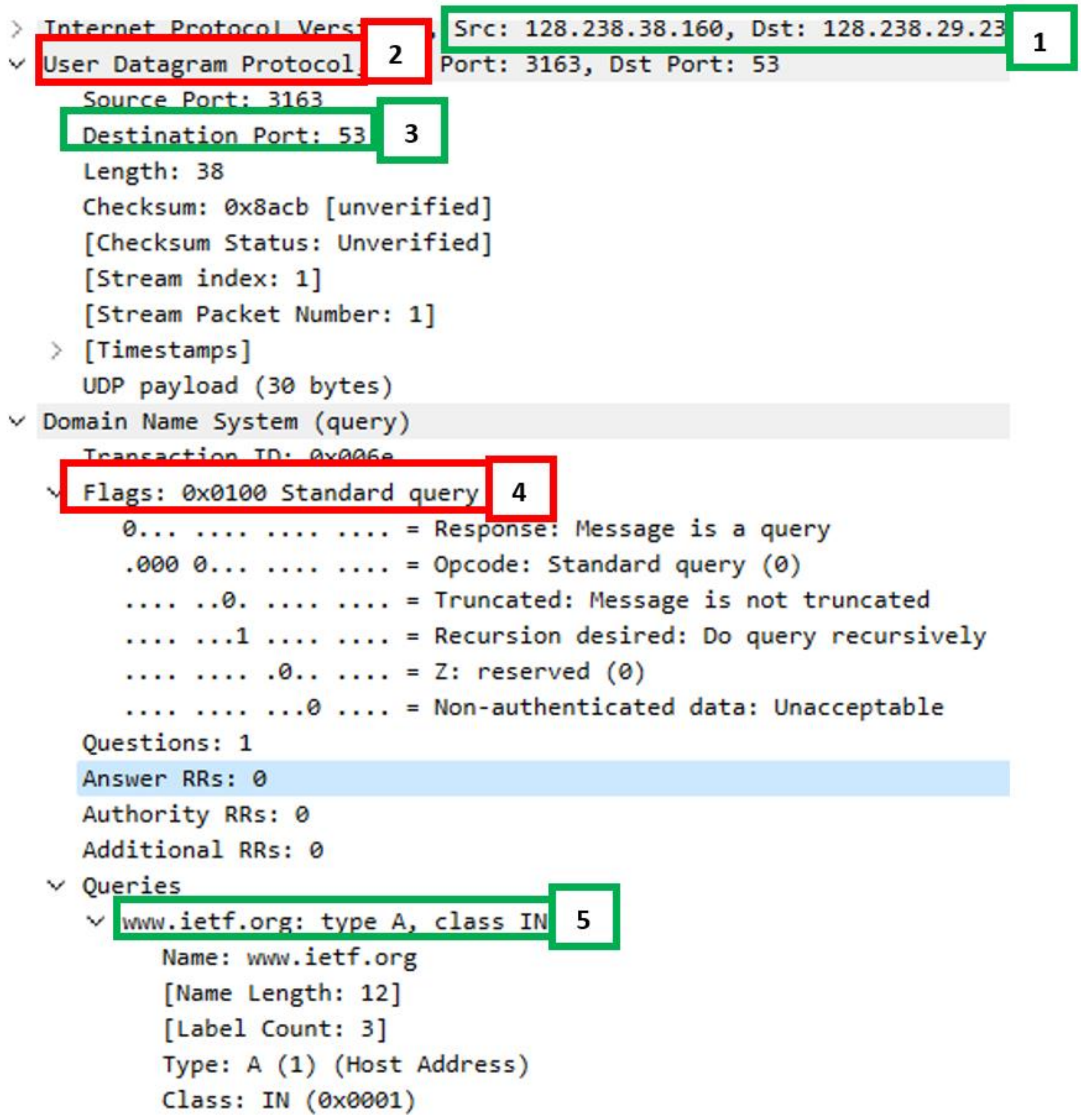
```
∨ Kerberos
  ∨ tgs-req
      pvno: 5
      msg-type: krb-tgs-req (12)
    ∨ padata: 1 item
      ∨ PA-DATA pA-TGS-REQ
        ∨ padata-type: pA-TGS-REQ (1)
          ∨ padata-value […]: 6e82041830820414a003020105a10302010ea2070
            ∨ ap-req
                pvno: 5
                msg-type: krb-ap-req (14)
                Padding: 0
              > ap-options: 00000000
              ∨ ticket
                  tkt-vno: 5
                  realm: DENYDC.COM
                ∨ sname
                    name-type: kRB5-NT-SRV-INST (2)
                  ∨ sname-string: 2 items
                      SNameString: krbtgt
                      SNameString: DENYDC.COM
                ∨ enc-part
                    etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
                    kvno: 2
                    cipher […]: 76873a46dedc5b7de4cd702aef30ae79cbd8
              ∨ authenticator
                  etype: eTYPE-DES-CBC-MD5 (3)
                  cipher […]: 60b1edf4e7433bef9d79f38bd2c9d2bb69ac030
    ∨ req-body
        Padding: 0
      > kdc-options: 40800000
        realm: DENYDC.COM
      ∨ sname
          name-type: kRB5-NT-SRV-HST (3)
        ∨ sname-string: 2 items
            SNameString: host
            SNameString: xp1.denydc.com
        till: Sep 13, 2037 03:48:05.000000000 GMT Summer Time
        nonce: 197296424
      ∨ etype: 7 items
          ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
          ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD (-133)
          ENCTYPE: eTYPE-ARCFOUR-MD4 (-128)
          ENCTYPE: eTYPE-DES-CBC-MD5 (3)
          ENCTYPE: eTYPE-DES-CBC-CRC (1)
          ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5-56 (24)
          ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)
```

Figure 11



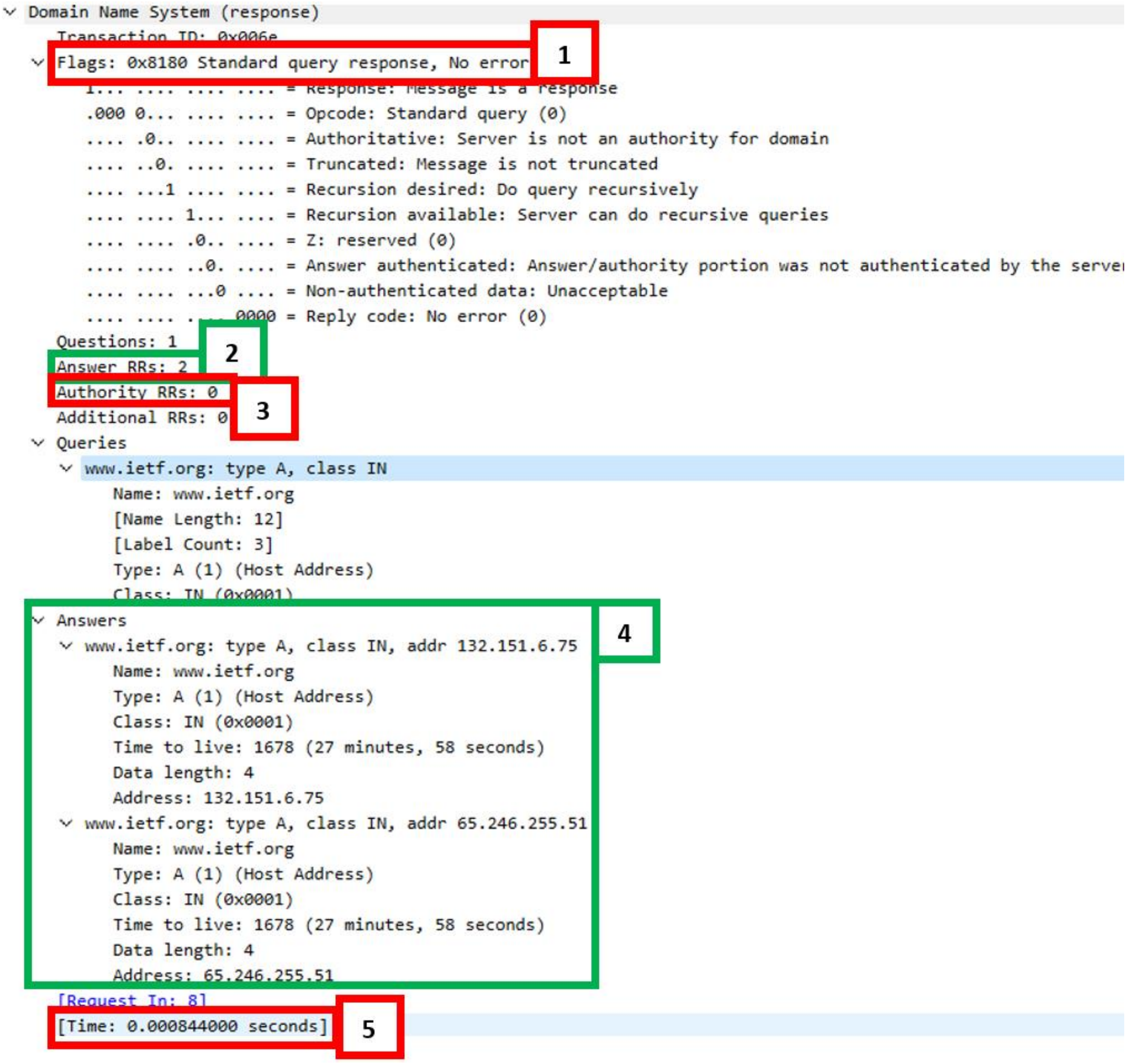
```
v Kerberos
  v tgs-rep
      pvno: 5
      msg-type: krb-tgs-rep (13)
      crealm: DENYDC.COM
    v cname
        name-type: kRB5-NT-PRINCIPAL (1)
      v cname-string: 1 item
          CNameString: des
    v ticket
        tkt-vno: 5
        realm: DENYDC.COM
      v sname
          name-type: kRB5-NT-SRV-HST (3)
        v sname-string: 2 items
            SNameString: host
            SNameString: xp1.denydc.com
      v enc-part
          etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
          kvno: 2
          cipher […]: e63bb88dd1d8f8b5aafe7b76e59e4f42e5e090b679e8a945
    v enc-part
        etype: eTYPE-DES-CBC-MD5 (3)
        cipher […]: 70e024fdb23293198556e63ca27554cf3dd36d0a548e9215906
    [Response to: 5]
    [Time from request: 0.000007000 seconds]
```

Figure 12



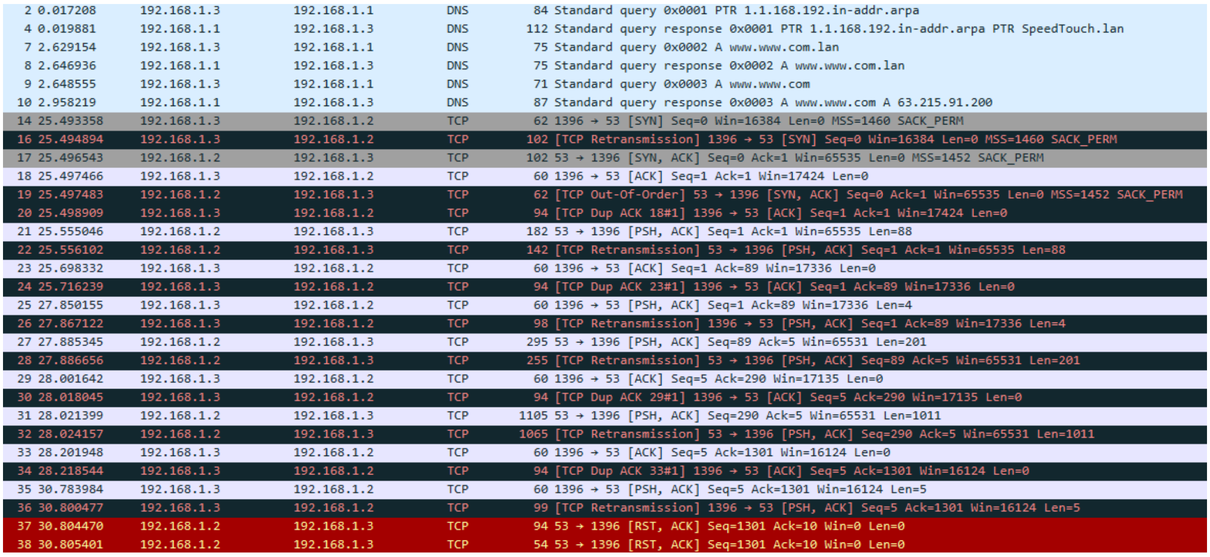
```
> Internet Protocol Versi   , Src: 128.238.38.160, Dst: 128.238.29.23
v User Datagram Protocol,  Port: 3163, Dst Port: 53
    Source Port: 3163
    Destination Port: 53
    Length: 38
    Checksum: 0x8acb [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
    [Stream Packet Number: 1]
  > [Timestamps]
    UDP payload (30 bytes)
v Domain Name System (query)
    Transaction ID: 0x006e
  v Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v www.ietf.org: type A, class IN
        Name: www.ietf.org
        [Name Length: 12]
        [Label Count: 3]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
```

Figure 13



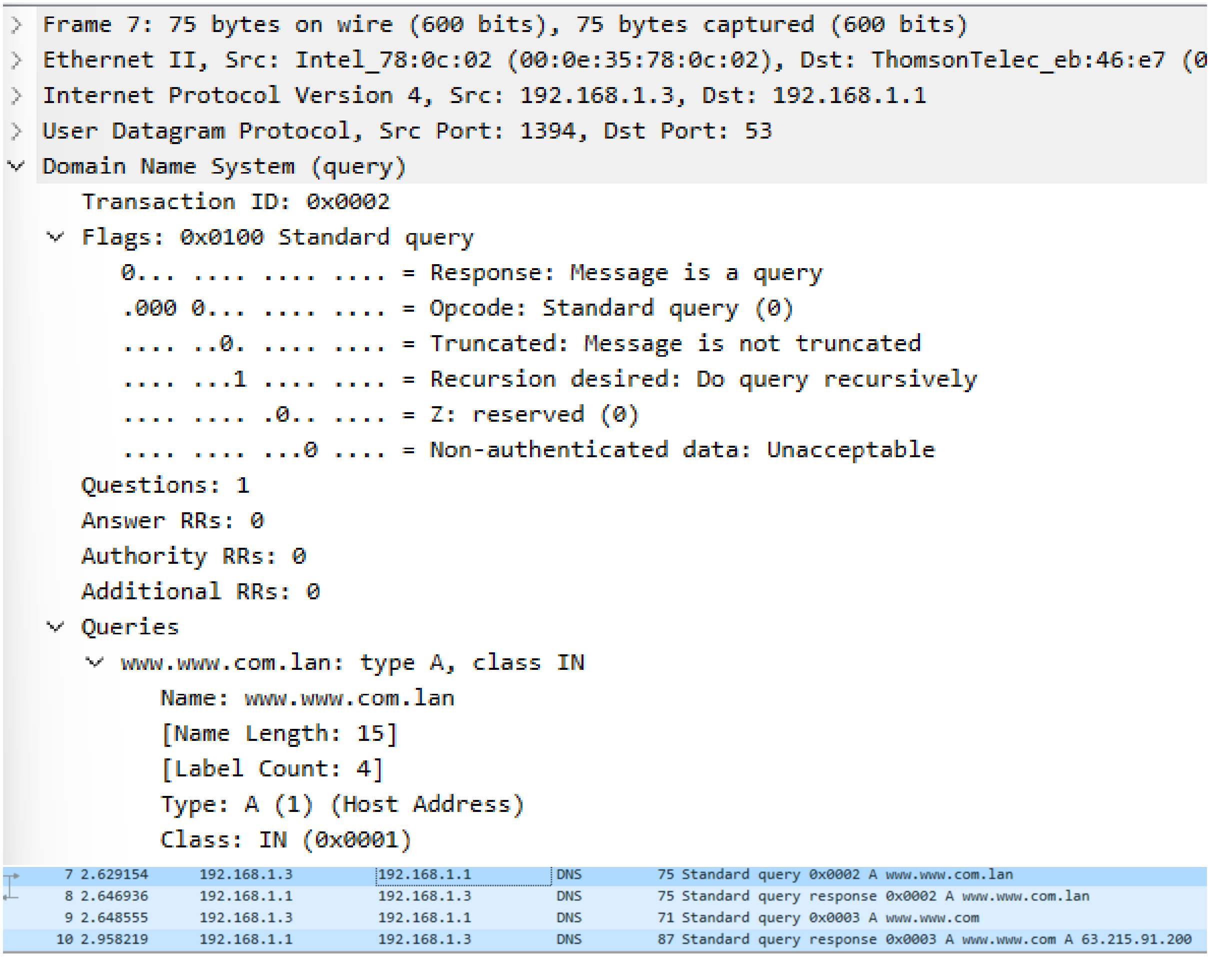
```
∨ Domain Name System (response)
    Transaction ID: 0x006e
  ∨ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for 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 serve
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  ∨ Queries
    ∨ www.ietf.org: type A, class IN
        Name: www.ietf.org
        [Name Length: 12]
        [Label Count: 3]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
  ∨ Answers
    ∨ www.ietf.org: type A, class IN, addr 132.151.6.75
        Name: www.ietf.org
        Type: A (1) (Host Address)
        Class: IN (0x0001)
        Time to live: 1678 (27 minutes, 58 seconds)
        Data length: 4
        Address: 132.151.6.75
    ∨ www.ietf.org: type A, class IN, addr 65.246.255.51
        Name: www.ietf.org
        Type: A (1) (Host Address)
        Class: IN (0x0001)
        Time to live: 1678 (27 minutes, 58 seconds)
        Data length: 4
        Address: 65.246.255.51
    [Request In: 8]
    [Time: 0.000844000 seconds]
```

Figure 14



```
 2 0.017208     192.168.1.3     192.168.1.1     DNS      84 Standard query 0x0001 PTR 1.1.168.192.in-addr.arpa
 4 0.019881     192.168.1.1     192.168.1.3     DNS     112 Standard query response 0x0001 PTR 1.1.168.192.in-addr.arpa PTR SpeedTouch.lan
 7 2.629154     192.168.1.3     192.168.1.1     DNS      75 Standard query 0x0002 A www.www.com.lan
 8 2.646936     192.168.1.1     192.168.1.3     DNS      75 Standard query response 0x0002 A www.www.com.lan
 9 2.648555     192.168.1.3     192.168.1.1     DNS      71 Standard query 0x0003 A www.www.com
10 2.958219     192.168.1.1     192.168.1.3     DNS      87 Standard query response 0x0003 A www.www.com A 63.215.91.200
14 25.493358    192.168.1.3     192.168.1.2     TCP      62 1396 → 53 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
16 25.494894    192.168.1.3     192.168.1.2     TCP     102 [TCP Retransmission] 1396 → 53 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
17 25.496543    192.168.1.2     192.168.1.3     TCP     102 53 → 1396 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1452 SACK_PERM
18 25.497466    192.168.1.3     192.168.1.2     TCP      60 1396 → 53 [ACK] Seq=1 Ack=1 Win=17424 Len=0
19 25.497483    192.168.1.2     192.168.1.3     TCP      62 [TCP Out-Of-Order] 53 → 1396 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1452 SACK_PERM
20 25.498909    192.168.1.3     192.168.1.2     TCP      94 [TCP Dup ACK 18#1] 1396 → 53 [ACK] Seq=1 Ack=1 Win=17424 Len=0
21 25.555046    192.168.1.2     192.168.1.3     TCP     182 53 → 1396 [PSH, ACK] Seq=1 Ack=1 Win=65535 Len=88
22 25.556102    192.168.1.2     192.168.1.3     TCP     142 [TCP Retransmission] 53 → 1396 [PSH, ACK] Seq=1 Ack=1 Win=65535 Len=88
23 25.698332    192.168.1.3     192.168.1.2     TCP      60 1396 → 53 [ACK] Seq=1 Ack=89 Win=17336 Len=0
24 25.716239    192.168.1.3     192.168.1.2     TCP      94 [TCP Dup ACK 23#1] 1396 → 53 [ACK] Seq=1 Ack=89 Win=17336 Len=0
25 27.850155    192.168.1.3     192.168.1.2     TCP      60 1396 → 53 [PSH, ACK] Seq=1 Ack=89 Win=17336 Len=4
26 27.867122    192.168.1.3     192.168.1.2     TCP      98 [TCP Retransmission] 1396 → 53 [PSH, ACK] Seq=1 Ack=89 Win=17336 Len=4
27 27.885345    192.168.1.2     192.168.1.3     TCP     295 53 → 1396 [PSH, ACK] Seq=89 Ack=5 Win=65531 Len=201
28 27.886656    192.168.1.2     192.168.1.3     TCP     255 [TCP Retransmission] 53 → 1396 [PSH, ACK] Seq=89 Ack=5 Win=65531 Len=201
29 28.001642    192.168.1.3     192.168.1.2     TCP      60 1396 → 53 [ACK] Seq=5 Ack=290 Win=17135 Len=0
30 28.018045    192.168.1.3     192.168.1.2     TCP      94 [TCP Dup ACK 29#1] 1396 → 53 [ACK] Seq=5 Ack=290 Win=17135 Len=0
31 28.021399    192.168.1.2     192.168.1.3     TCP    1105 53 → 1396 [PSH, ACK] Seq=290 Ack=5 Win=65531 Len=1011
32 28.024157    192.168.1.2     192.168.1.3     TCP    1065 [TCP Retransmission] 53 → 1396 [PSH, ACK] Seq=290 Ack=5 Win=65531 Len=1011
33 28.201948    192.168.1.3     192.168.1.2     TCP      60 1396 → 53 [ACK] Seq=5 Ack=1301 Win=16124 Len=0
34 28.218544    192.168.1.3     192.168.1.2     TCP      94 [TCP Dup ACK 33#1] 1396 → 53 [ACK] Seq=5 Ack=1301 Win=16124 Len=0
35 30.783984    192.168.1.3     192.168.1.2     TCP      60 1396 → 53 [PSH, ACK] Seq=5 Ack=1301 Win=16124 Len=5
36 30.800477    192.168.1.3     192.168.1.2     TCP      99 [TCP Retransmission] 1396 → 53 [PSH, ACK] Seq=5 Ack=1301 Win=16124 Len=5
37 30.804470    192.168.1.2     192.168.1.3     TCP      94 53 → 1396 [RST, ACK] Seq=1301 Ack=10 Win=0 Len=0
38 30.805401    192.168.1.2     192.168.1.3     TCP      54 53 → 1396 [RST, ACK] Seq=1301 Ack=10 Win=0 Len=0
```

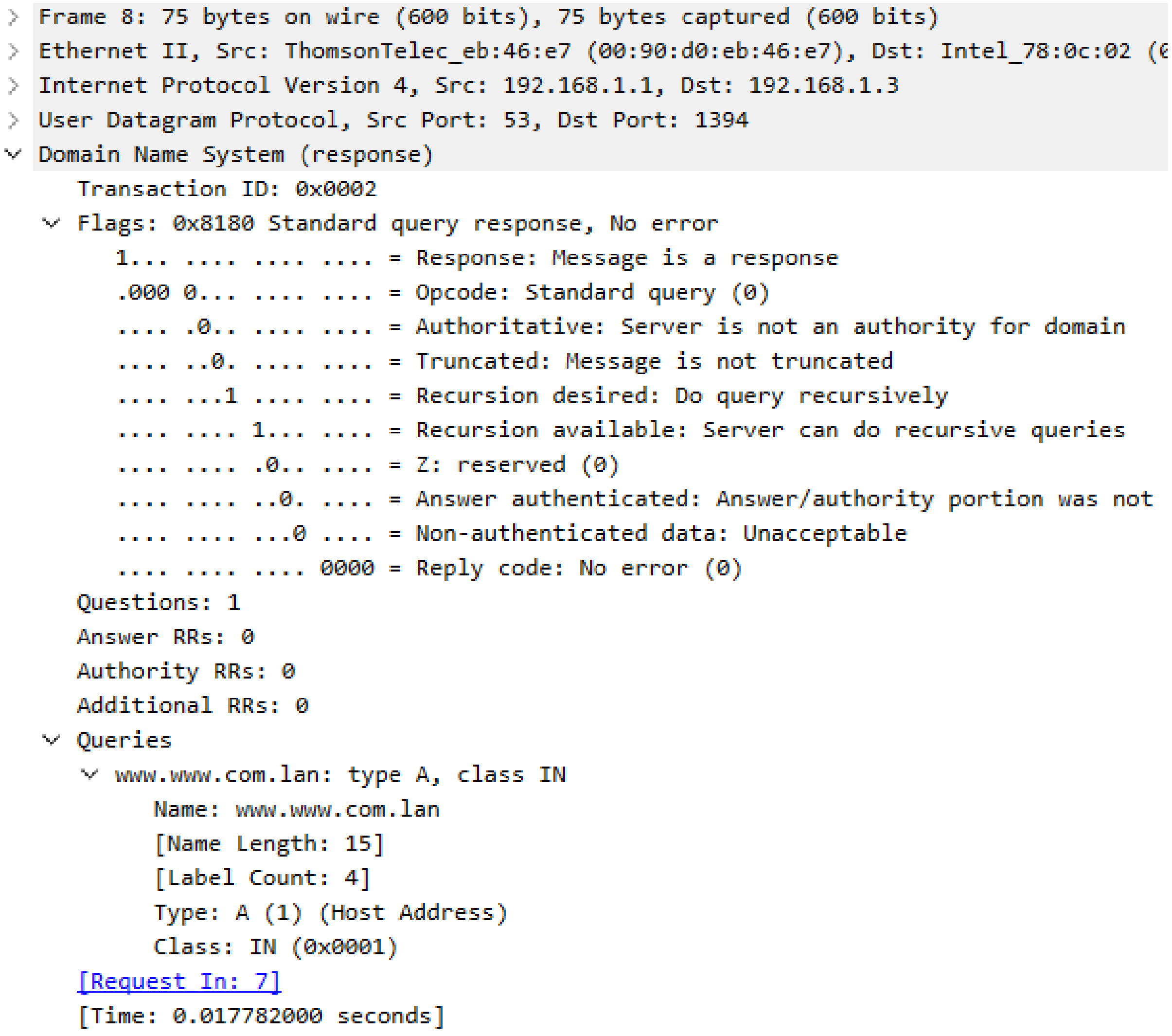
Figure 15



```
> Frame 7: 75 bytes on wire (600 bits), 75 bytes captured (600 bits)
> Ethernet II, Src: Intel_78:0c:02 (00:0e:35:78:0c:02), Dst: ThomsonTelec_eb:46:e7 (0
> Internet Protocol Version 4, Src: 192.168.1.3, Dst: 192.168.1.1
> User Datagram Protocol, Src Port: 1394, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x0002
  v Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v www.www.com.lan: type A, class IN
        Name: www.www.com.lan
        [Name Length: 15]
        [Label Count: 4]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
```

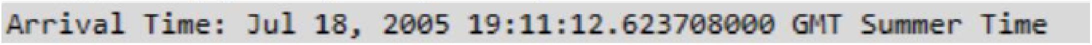
```
 7 2.629154     192.168.1.3     192.168.1.1     DNS     75 Standard query 0x0002 A www.www.com.lan
 8 2.646936     192.168.1.1     192.168.1.3     DNS     75 Standard query response 0x0002 A www.www.com.lan
 9 2.648555     192.168.1.3     192.168.1.1     DNS     71 Standard query 0x0003 A www.www.com
10 2.958219     192.168.1.1     192.168.1.3     DNS     87 Standard query response 0x0003 A www.www.com A 63.215.91.200
```

Figure 16



```
> Frame 8: 75 bytes on wire (600 bits), 75 bytes captured (600 bits)
> Ethernet II, Src: ThomsonTelec_eb:46:e7 (00:90:d0:eb:46:e7), Dst: Intel_78:0c:02 (0
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.3
> User Datagram Protocol, Src Port: 53, Dst Port: 1394
v Domain Name System (response)
     Transaction ID: 0x0002
   v 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
        .... .... ...0 .... = Non-authenticated data: Unacceptable
        .... .... .... 0000 = Reply code: No error (0)
     Questions: 1
     Answer RRs: 0
     Authority RRs: 0
     Additional RRs: 0
   v Queries
      v www.www.com.lan: type A, class IN
           Name: www.www.com.lan
           [Name Length: 15]
           [Label Count: 4]
           Type: A (1) (Host Address)
           Class: IN (0x0001)
     [Request In: 7]
     [Time: 0.017782000 seconds]
```

Figure 17



```
Arrival Time: Jul 18, 2005 19:11:12.623708000 GMT Summer Time
```

Figure 18