



Nottingham Trent  
University

COMP40751  
Information and Computer Security  
Secure Protocols Portfolio  
N1334679 – Karunakar Reddy Machupalli

**Table of Contents**

<b>1.Http protocol/TCP Handshake.....</b>	<b>06</b>
---	-----------

1.1 Http/TCP Handshake.....	06
1.2 In-Depth Analysis of Http Traces.....	06
1.2.1 SYN (Synchronize).....	06
1.2.1.1 In-Depth Analysis of SYN Packet from Http Trace 1.....	06
1.2.1.2 Summary.....	06
1.2.2 SYN – ACK (Synchronize-Acknowledge).....	08
1.2.2.1 In-Depth Analysis of SYN-ACK Packet from Http Trace 1.....	08
1.2.2.2 Summary.....	09
1.2.3 ACK (Acknowledge).....	09
1.2.3.1 In-Depth Analysis of ACK Packet from Http Trace 1.....	09
1.2.3.2 Summary.....	10
1.2.4 Http Trace File 2 Analysis.....	11
1.2.5 Http Trace File 3 Analysis.....	12
1.3 Network Parameters and Their Significance.....	12
1.4 Authentication.....	13
1.4.1 Initial Request.....	13
1.4.2 In-Depth Analysis of Frame 6 from Http Trace 3.....	13
1.4.3 Server Response.....	14
1.4.4 In-Depth Analysis of Frame 9 from Http Trace 3.....	14
1.4.5 Client Resends Request.....	15
1.4.6 In-Depth Analysis of Frame 65 from Http Trace 3.....	15
1.4.7 Server Response – Authenticated.....	16
1.4.8 In-Depth Analysis of Frame 68 from Http Trace 3.....	16
1.4.9 Key Headers and Tokens.....	17
1.5 Common Vulnerabilities and Security Measures In HTTP/TCP Handshake.....	17
1.5.1 SYN Flooding Attack.....	17
1.5.1.1 Impact.....	18
1.5.1.2 Mitigation.....	18
1.5.2 IP Spoofing Attack.....	18

1.5.2.1 Impact.....	18
1.5.2.2 Mitigation.....	19
2. SSL/TLS Cipher Suites.....	19
2.1 SSL/TLS Handshake.....	19
2.2 In-Depth Comparative Evaluation of SSL Trace.....	20
2.3 In-Depth Security Aspects.....	23
2.3.1 Chosen Cipher Suites.....	23
2.3.2 Security Strength Analysis.....	24
2.3.3 Justification for Choosing TLS_RSA_WITH_RC4_128_MD5 Cipher Suite.....	24
2.3.4 Modern Security Practices.....	25
3. Kerberos.....	25
3.1 Kerberos.....	25
3.2 In-Depth Analysis of Kerberos Traces.....	26
3.2.1 AS-REQ (Authentication Service Request).....	26
3.2.2 AS-REP (Authentication Service Reply).....	27
3.2.3 TGS-REQ (Ticket Granting Service Request).....	28
3.2.4 TGS-REP (Ticket Granting Service Reply).....	29
3.3 In-Depth Security Features.....	30
3.3.1 Algorithms.....	30
3.3.2 Security Strength.....	31
3.3.3 Kerberos Limitations.....	31
4. DNS Anomalies.....	32
4.1 DNS Protocol.....	32
4.2 In-Depth Analysis of DNS Trace 1.....	32

4.2.1 DNS Request.....	32
4.2.2 DNS Response.....	33
4.3 Comparative Evaluation of Normal DNS and Anomalous DNS Pair.....	34
4.4 Normal DNS Pair.....	35
4.4.1 Query.....	35
4.4.2 Response.....	35
4.5 Anomalous DNS Pair.....	36
4.5.1 Query.....	36
4.5.2 Response.....	37
4.6 Nature and Detection of Anomaly.....	38
4.7 DNS security limitations.....	38
5. References.....	38

## Section 1: Http Protocol/TCP Handshake

### **1.1 Http/TCP Handshake**

The TCP handshaking mechanism, referred to as the three-way handshake, is an essential procedure for establishing a reliable connection between a client and a server. This procedure guarantees that both parties are prepared to talk and can align their sequence numbers. The procedure begins with the client transmitting a SYN (synchronize) packet to the server. This packet signifies the client's intent to initiate a connection and contains the beginning sequence number chosen by the client. The SYN packet serves as a request to initiate communication and synchronize sequence numbers. The sequence number is a randomly selected beginning value employed to monitor the order of the packets. Upon receiving the SYN packet, the

server replies with a SYN-ACK (synchronize-acknowledge) packet. This packet confirms the reception of the SYN packet and contains the server's initial sequence number. The SYN-ACK packet serves two functions: it acknowledges the client's SYN packet by adjusting the acknowledgement number to the client's sequence number incremented by one, and additionally it includes the server's beginning sequence number. The client transmits an ACK (acknowledge) packet to the server, therefore concluding the three-way handshake. This packet confirms the receipt of the server's SYN-ACK packet. The ACK packet verifies that the client has received the server's SYN-ACK packet by incrementing the server's sequence number by one in the acknowledgement number. The handshake is finalized, and the link is created.

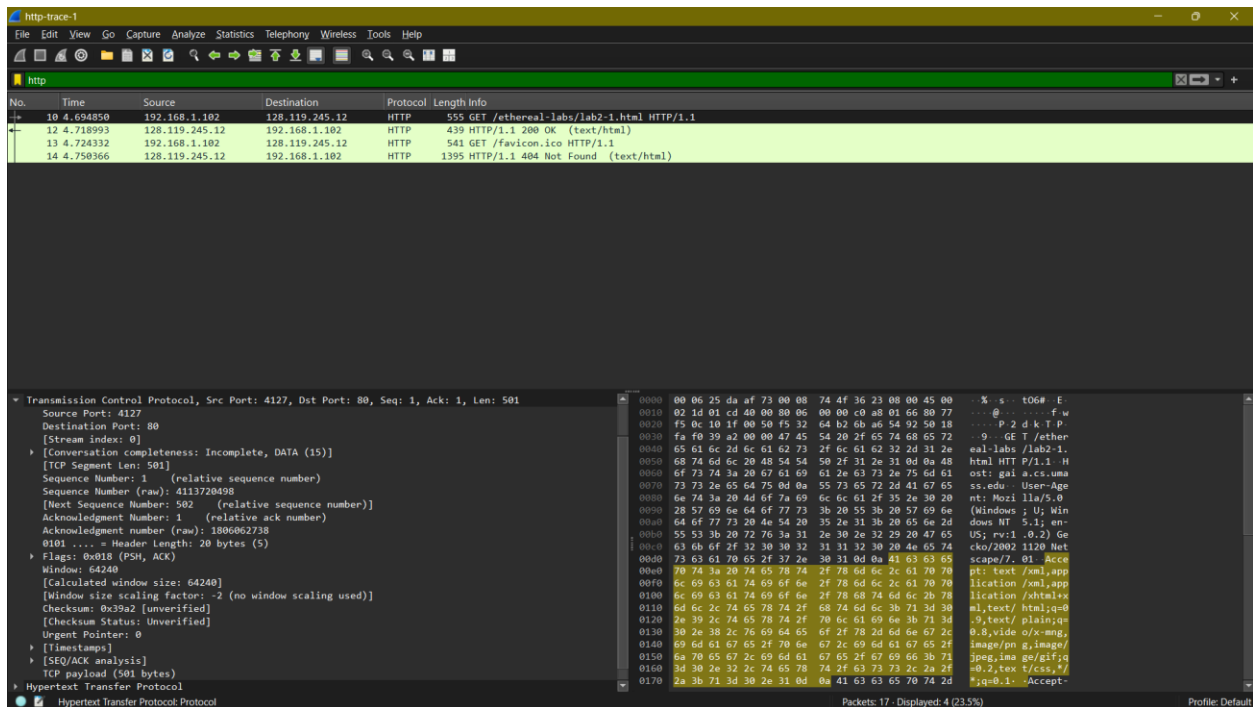
## 1.2 In-Depth Analysis of Http Traces

### 1.2.1 SYN (Synchronize)

The client establishes the connection by transmitting a TCP packet with the SYN flag activated. This stage establishes the initial sequence number (ISN) and signifies the client's intention to initiate a connection.

#### 1.2.1.1 In-Depth Analysis of SYN Packet from Http Trace 1

Field	Details
Frame Number	10
Time	7.236929 seconds
Source IP Address	192.168.1.102
Destination IP Address	128.119.245.12
Source Port	4307
Destination Port	80
Flags	0x018 (PSH, ACK)
PSH (Push)	Set
ACK (Acknowledgment)	Set
SYN (Synchronize)	Present
Sequence Number	1 (relative sequence number)
Sequence Number (raw)	4246551714
Window Size	64240



### 1.2.1.2 Summary

This packet constitutes an aspect of the TCP three-way handshake process, specifically the SYN packet dispatched by the client to start a connection with the server. The SYN flag is present, signifying the initiation of the connection. The sequence number is established at 1, and the window size is 64240, indicating the volume of data the sender is prepared to accept prior to requiring an acknowledgement.

### 1.2.2 SYN – ACK (Synchronize-Acknowledge)

The server replies with a TCP message containing both the SYN and ACK flags activated. This stage acknowledges the client's SYN and delivers the server's starting sequence number.

#### 1.2.2.1 In-Depth Analysis of SYN-ACK Packet from Http Trace 1

Field	Details
Frame Number	12

<b>Time</b>	<b>7.260813 seconds</b>
<b>Source IP Address</b>	<b>128.119.245.12</b>
<b>Destination IP Address</b>	<b>192.168.1.102</b>
<b>Source Port</b>	<b>80</b>
<b>Destination Port</b>	<b>4307</b>
<b>Flags</b>	<b>0x018 (PSH, ACK)</b>
<b>PSH (Push)</b>	<b>Set</b>
<b>ACK (Acknowledgment)</b>	<b>Set</b>
<b>SYN (Synchronize)</b>	<b>Present</b>
<b>Acknowledgment Number</b>	<b>502 (relative acknowledgment number)</b>
<b>Acknowledgment Number (raw)</b>	<b>4246552215</b>
<b>Window Size</b>	<b>6432</b>

The image shows a Wireshark packet capture window titled "Wireshark - Packet 12 - http-trace-1". The packet list on the left shows "Frame 12: 439 bytes on wire (3512 bits), 439 bytes captured (3512 bits)". The packet details pane on the right shows the following information:

- Ethernet II**, Src: LinksysGroup.daaaf:73 (00:06:25:daaaf:73), Dst: Dell\_4f:36:23 (00:08:74:4f:36:23)
- Internet Protocol Version 4**, Src: 128.119.245.12, Dst: 192.168.1.102
- Transmission Control Protocol**, Src Port: 80, Dst Port: 4127, Seq: 1, Ack: 502, Len: 385
  - Source Port: 80
  - Destination Port: 4127
  - [Stream index: 0]
  - [Conversation completeness: Incomplete, DATA (15)]
  - [TCP Segment Len: 385]
  - Sequence Number: 1 (relative sequence number)
  - Sequence Number (raw): 1506062738
  - [Next Sequence Number: 386 (relative sequence number)]
  - Acknowledgment Number: 502 (relative ack number)
  - Acknowledgment number (raw): 4113720999
  - 0101 .... = Header Length: 20 bytes (5)
  - Flags: 0x018 (PSH, ACK)
  - Window: 6432
  - [Calculated window size: 6432]
  - [Window size scaling factor: -2 (no window scaling used)]
  - Checksum: 0x7alc [unverified]
  - [Checksum Status: Unverified]
  - Urgent Pointer: 0
  - [Timestamps]
  - [SEQ/ACK analysis]
  - TCP payload (385 bytes)
- Hypertext Transfer Protocol**
- Line-based text data: text/html (3 lines)

The status bar at the bottom shows: "No. 12 - Time: 4.718993 - Source: 128.119.245.12 - Destination: 192.168.1.102 - Protocol: HTTP - Length: 439 - Info: HTTP/1.1 200 OK (text/html)".

### 1.2.2.2 Summary

This packet contains a component of the TCP three-way handshake process, specifically the SYN-ACK packet transmitted by the server in reply to the client's SYN packet. The presence of the SYN and ACK flags signifies the server's acknowledgement of the client's connection request. The acknowledgement number is established at 502, and the window size is 6432, indicating the volume of data the server is prepared to accept prior to requiring an acknowledgement.

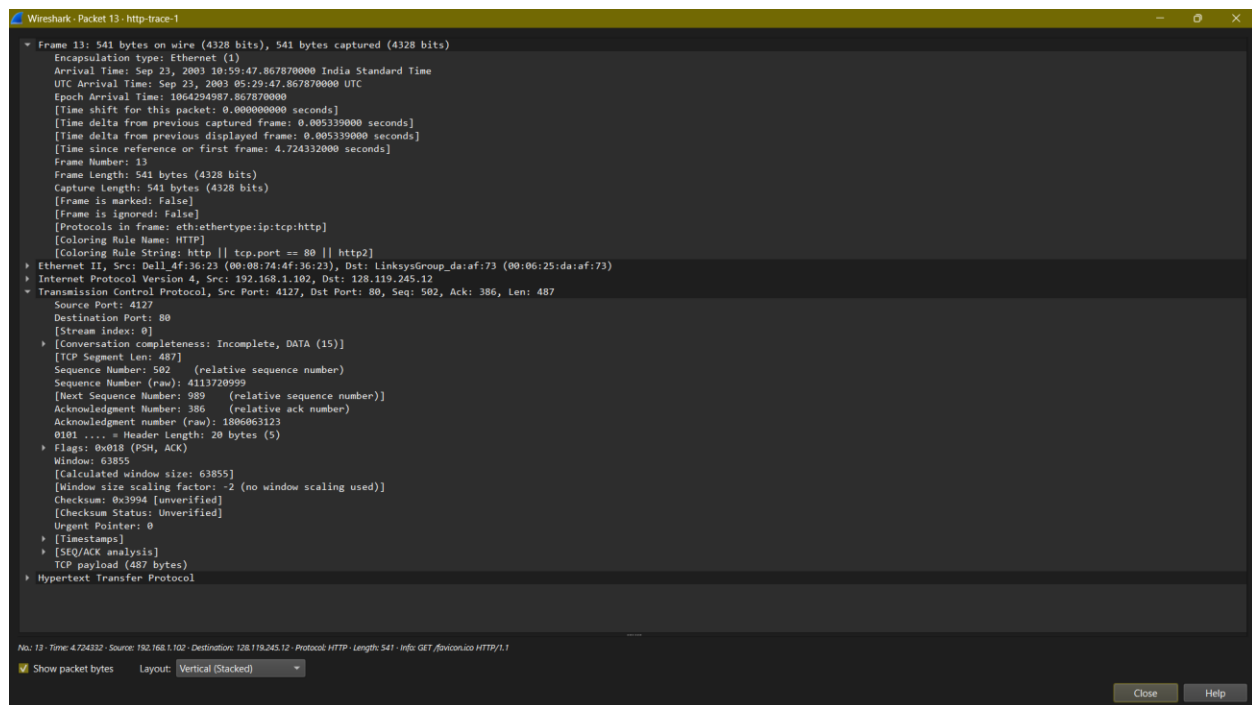


### 1.2.3 ACK (Acknowledge)

The client transmits a concluding TCP packet with the ACK flag activated. This stage acknowledges the server's SYN-ACK, finishing the handshake and establishing a dependable connection.

#### 1.2.3.1 In-Depth Analysis of ACK Packet from Http Trace 1

Field	Details
Frame Number	13
Time	7.305485 seconds
Source IP Address	192.168.1.102
Destination IP Address	128.119.245.12
Source Port	4307
Destination Port	80
Flags	0x018 (PSH, ACK)
PSH (Push)	Set
ACK (Acknowledgment)	Set
Sequence Number	502 (relative sequence number)
Sequence Number (raw)	4246552215
Acknowledgment Number	1004 (relative acknowledgment number)
Acknowledgment Number (raw)	2354562744



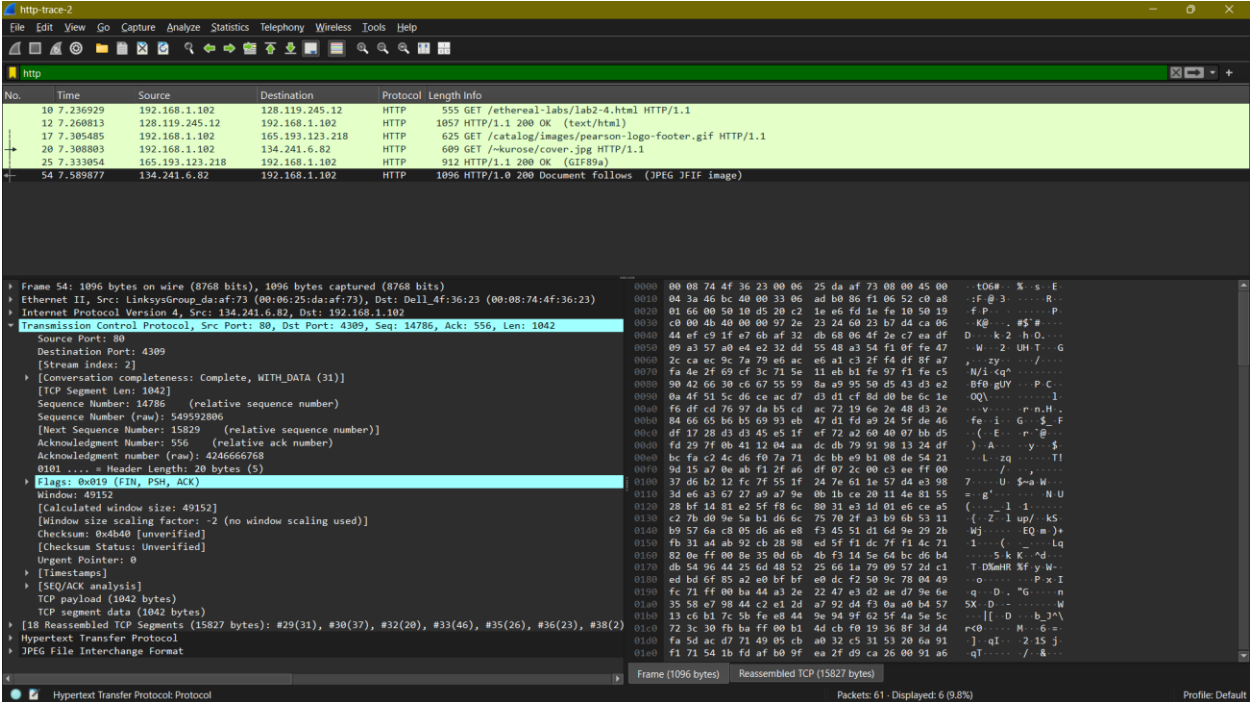
### 1.2.3.2 Summary

This packet contains the concluding segment of the TCP three-way handshake process, particularly the ACK packet dispatched by the client to confirm receipt of the server's SYN-ACK packet. The ACK flag signifies the client's acknowledgement of the server's answer. The sequence number is established at 502, and the acknowledgement number is set at 1004, therefore completing the handshake and establishing a connection between the client and server.

### 1.2.4 Http Trace File 2 Analysis

Packet Type	Source IP	Destination IP	Source Port	Destination Port	Flags	Description
SYN Packet	192.168.1.102	128.119.245.12	4307	80	SYN	The client dispatched a SYN message to commence the handshake procedure.
SYN-ACK Packet	128.119.245.12	192.168.1.102	-	-	SYN, ACK	The server acknowledged the SYN request and suggested its own sequence number.

ACK Packet	192.168.1.102	128.119.245.12	-	-	ACK	The client concluded the handshake by transmitting an acknowledgement, so enabling HTTP communication to continue.
------------	---------------	----------------	---	---	-----	--



1.2.5 Http Trace File 3 Analysis

Packet Type	Source IP	Destination IP	Source Port	Destination Port	Flags	Description
SYN Packet	192.168.1.102	128.119.245.12	4335	80	SYN	The client initiated a connection using a dynamic source port.
SYN-ACK Packet	128.119.245.12	192.168.1.102	-	-	SYN, ACK	The server responded, signaling its readiness to establish a connection.
ACK Packet	192.168.1.102	128.119.245.12	-	-	ACK	The client sent the final acknowledgment, successfully establishing the session.

### 1.3 Network Parameters and Their Significance

The network parameters significant to the TCP handshaking procedure includes IP addresses, port numbers, sequence numbers, and acknowledgement numbers. IP addresses specify the source and destination devices within a network, while port numbers identify the activities or services on them. Sequence numbers guarantee the accurate transmission order of data and assist in identifying lost packets. Acknowledgement numbers verify the receipt of packets and facilitate the regulation of data flow.

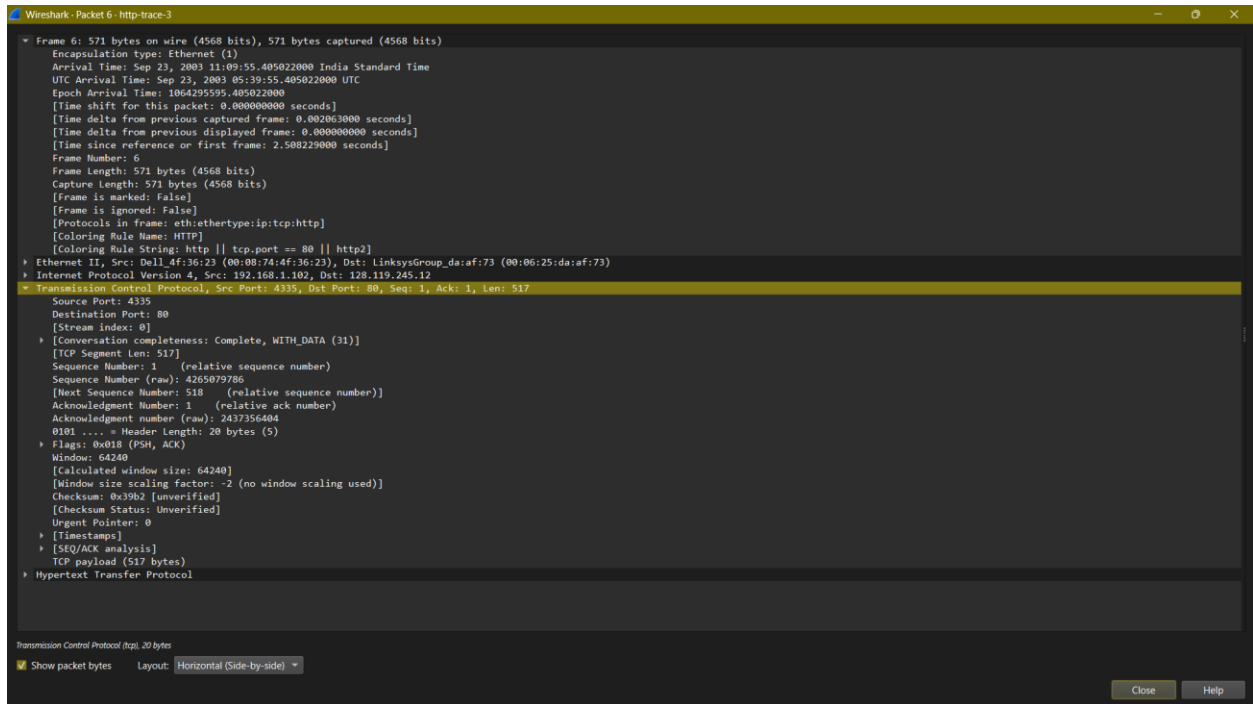
### 1.4 Authentication

#### 1.4.1 Initial Request

The client (192.168.1.102) transmits an HTTP GET request to the server (128.119.245.12) for the resource /ethereal-labs/protected\_pages/lab2-5.html.

#### 1.4.2 In-Depth Analysis of Frame 6 from Http Trace 3

Field	Value
Method	GET
Path	/ethereal-labs/protected_pages/lab2-5.html
HTTP Version	HTTP/1.1
Host	gaia.cs.umass.edu
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120 Netscape/7.01
Accept	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,/*;q=0.1
Accept-Language	en-us, en;q=0.50
Accept-Encoding	gzip, deflate, compress;q=0.9
Accept-Charset	ISO-8859-1, utf-8;q=0.66, /*;q=0.66
Keep-Alive	300
Connection	keep-alive



### 1.4.3 Server Response

The server returns a 401 Authorisation Required response, signifying that authentication is necessary to access the resource.

### 1.4.4 In-Depth Analysis of Frame 9 from Http Trace 3

Field	Value
HTTP Version	HTTP/1.1
Status Code	401 Authorization Required
Date	Tue, 23 Sep 2003 05:29:50 GMT
Server	Apache/2.0.40 (Red Hat Linux)
WWW-Authenticate	Basic realm="Ethereal Labs"
Content-Length	401
Keep-Alive	timeout=10, max=100
Connection	Keep-Alive
Content-Type	text/html; charset=ISO-8859-1



### 1.4.6 In-Depth Analysis of Frame 65 from Http Trace 3

Field	Value
Method	GET
Path	/ethereal-labs/protected_pages/lab2-5.html
HTTP Version	HTTP/1.1
Host	gaia.cs.umass.edu
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120 Netscape/7.01
Accept	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,/;q=0.1
Accept-Language	en-us, en;q=0.50
Accept-Encoding	gzip, deflate, compress;q=0.9
Accept-Charset	ISO-8859-1, utf-8;q=0.66, *;q=0.66
Keep-Alive	300
Connection	keep-alive

Authorization	Basic dXNlcm5hbWU6cGFzc3dvcmQ=
---------------	--------------------------------

```

Wireshark - Packet 65 - http-trace-3
  Frame 65: 622 bytes on wire (4976 bits), 622 bytes captured (4976 bits)
  Ethernet II, Src: Dell 4f:36:23 (00:08:74:4f:36:23), Dst: LinksysGroup_da:af:73 (00:06:25:da:af:73)
  Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
  Transmission Control Protocol, Src Port: 4342, Dst Port: 80, Seq: 1, Ack: 1, Len: 568
    Source Port: 4342
    Destination Port: 80
    [Stream index: 2]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 568]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 4269150442
    [Next Sequence Number: 569 (relative sequence number)]
    Acknowledgment Number: 1 (relative ack number)
    Acknowledgment number (raw): 2462372174
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 64240
    [calculated window size: 64240]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x39e5 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    [Timestamps]
    [SEQ/ACK analysis]
    TCP payload (568 bytes)
  Hypertext Transfer Protocol
    GET /etheral-labs/protected_pages/lab2-5.html HTTP/1.1\r\n
    Host: gala.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120 Netscape/7.01\r\n
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1\r\n
    Accept-Language: en-us,en;q=0.50\r\n
    Accept-Encoding: gzip, deflate, compress;q=0.9\r\n
    Accept-Charset: ISO-8859-1, utf-8;q=0.66,*;q=0.66\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Authorization: Basic ZXNlcm5hbWU6cGFzc3dvcmQ=\r\n
    \r\n
    [Reasons in frame: 68]
    [Full request URI: http://gala.cs.umass.edu/etheral-labs/protected_pages/lab2-5.html]
  HTTP Authorization header (http.authorization), 51 bytes
  Show packet bytes Layout: Horizontal (Side-by-side)
  Close Help

```

### 1.4.7 Server Response - Authenticated

The server returns a 200 OK status, signifying that authentication was successful and the requested resource is delivered.

### 1.4.8 In-Depth Analysis of Frame 68 from Http Trace 3

Field	Value
HTTP Version	HTTP/1.1
Status Code	200 OK
Date	Tue, 23 Sep 2003 05:29:50 GMT
Server	Apache/2.0.40 (Red Hat Linux)
Last-Modified	Tue, 23 Sep 2003 05:29:00 GMT
ETag	"1bfed-49-79d5bf00"
Accept-Ranges	bytes
Content-Length	73

Keep-Alive	timeout=10, max=100
Connection	Keep-Alive
Content-Type	text/html; charset=ISO-8859-1

```

Ethernet II, Src: LinksysGroup_dasaf:73 (00:06:25:da:af:73), Dst: Dell_4f:36:23 (00:08:74:4f:36:23)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102
Transmission Control Protocol, Src Port: 80, Dst Port: 4342, Seq: 1, Ack: 569, Len: 445
  Source Port: 80
  Destination Port: 4342
  [Stream index: 2]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 445]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 2462372174
  [Next Sequence Number: 446 (relative sequence number)]
  Acknowledgment Number: 569 (relative ack number)
  Acknowledgment number (raw): 4269151010
  0101 .... = Header Length: 20 bytes (5)
  Flags: 0x018 (PSH, ACK)
  Window: 6816
  [Calculated window size: 6816]
  [Window size scaling factor: -2 (no window scaling used)]
  Checksum: 0x6020 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  [Timestamps]
  [SYN/ACK analysis]
  TCP payload (445 bytes)
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
    Date: Tue, 23 Sep 2003 05:40:14 GMT\r\n
    Server: Apache/2.0.40 (Red Hat Linux)\r\n
    Last-Modified: Tue, 23 Sep 2003 04:03:59 GMT\r\n
    ETag: "626ec-84-49caa9e0"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 132\r\n
    [Content length: 132]
    Keep-Alive: timeout=10, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    \r\n
    [Request in frame: 65]
    [Time since request: 0.024878000 seconds]
    [Request URI: /etherreal-labs/protected_pages/lab2-5.html]
    [Full request: GET http://paas.cs.umd.edu/etherreal-labs/protected_pages/lab2-5.html]
    File Data: 132 bytes
  HTTP Response Status Code (http.response.code): 3 bytes
  Show packet bytes
  Layout: Horizontal (Side-by-side)
  Close Help

```

## 1.4.9 Key Headers and Tokens

**WWW-Authenticate:** This header is transmitted by the server to signify that the client must provide authentication to access the requested resource. This indicates Basic authentication with the realm "Ethereal Labs".

**Authorization:** This header is transmitted by the client in the following request to provide the credentials. The credentials are encoded in Base64.

## 1.5 Common Vulnerabilities and Security Measures In HTTP/TCP Handshake

### 1.5.1 SYN Flooding Attack

#### 1.5.1.1 Impact



A SYN flood attack uses the TCP handshake process to overwhelm a target server. The attacker floods the target server with SYN packets. Each SYN packet requests a new TCP connection. The server sends a SYN-ACK packet to each SYN packet to indicate its readiness to connect. The attacker refuses to transmit the final ACK packet to finish the handshake. This leaves the connection half-open. Memory and processing power are allocated for each half-open connection by the server. As half-open connections increase, server resources are depleted. Eventually, the server cannot handle any more connections, even authorised ones. The server is too busy to reply to valid users, denying service.

#### **1.5.1.2 Mitigation**

SYN Cookies encodes the connection state into the SYN-ACK packet sequence number. The server can validate the sequence number in the client's ACK packet to confirm the connection request. This reduces half-open connection resource allocation. The backlog queue holds half-open connections. By expanding this queue, the server may tolerate more half-open connections before overloading. This is a temporary fix that doesn't fix the attack. Firewalls and IDS can block malicious SYN packets. These systems can detect SYN flood patterns and prevent questionable traffic. Rate restriction can limit the number of SYN packets a server accepts from one IP address over time. This can prevent an attacker from flooding the server with SYN packets. Load balancers send traffic to numerous servers, limiting the impact of a SYN flood assault on one. This allows legal traffic to be processed even if one server is attacked.

### **1.5.2 IP Spoofing Attack**

#### **1.5.2.1 Impact**

IP spoofing allows attackers to hide by changing the packet header source IP address. An attacker transmits packets with a faked source IP address to appear to be from a trusted source. These faked packets reach the target via the network. Since the originating IP address is spoofed, the destination thinks the packets are authentic. IP spoofing can be used for DoS, MitM, and Session Hijacking attacks.

#### **1.5.2.2 Mitigation**

Network gateway packet filtering can detect and block suspicious or conflicting source IP addresses. Deep packet inspection detects irregularities through analysis of packet contents. To verify packet source IP addresses, network administrators might configure routers to filter

incoming and outgoing packets. Strong authentication and data encryption can help secure communication and verify packets. IPsec provides end-to-end security. Continuous network traffic monitoring and anomaly detection systems can detect IP spoofing trends. Rate limiting can reduce faked packet floods by restricting the amount of packets allowed from a single IP address over time.

## Section 2: SSL/TLS Cipher Suites

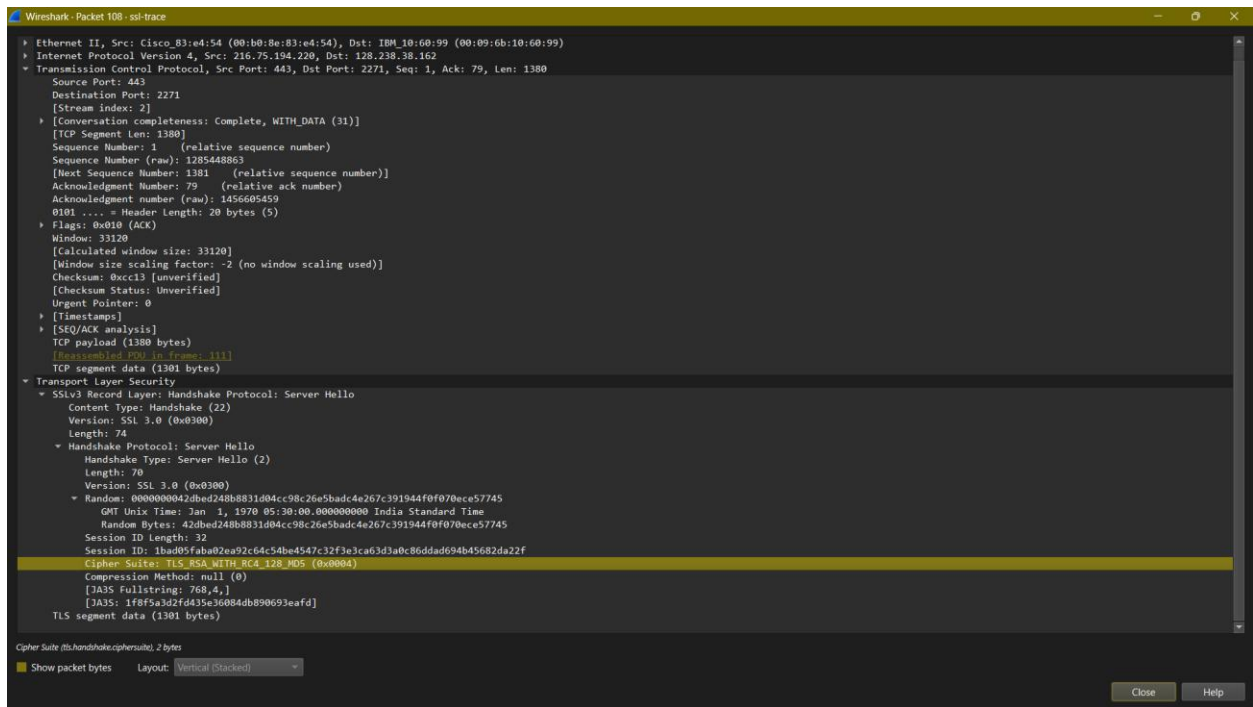
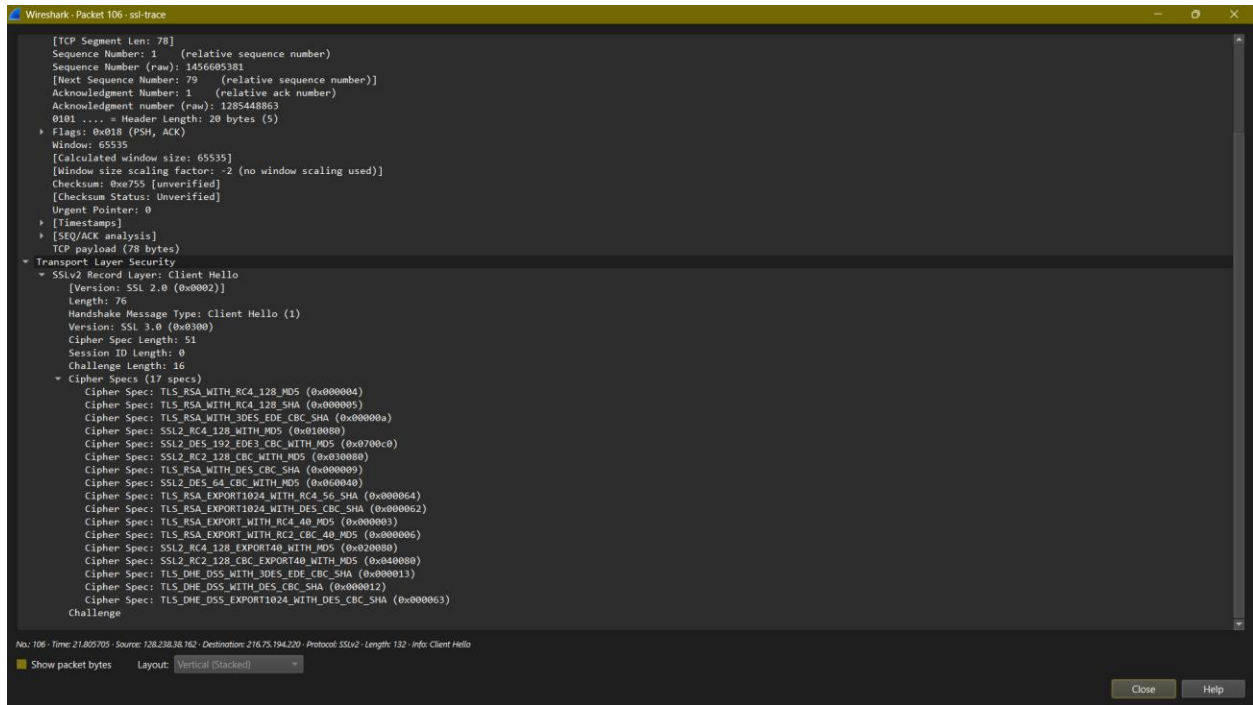
### 2.1 SSL/TLS Handshake

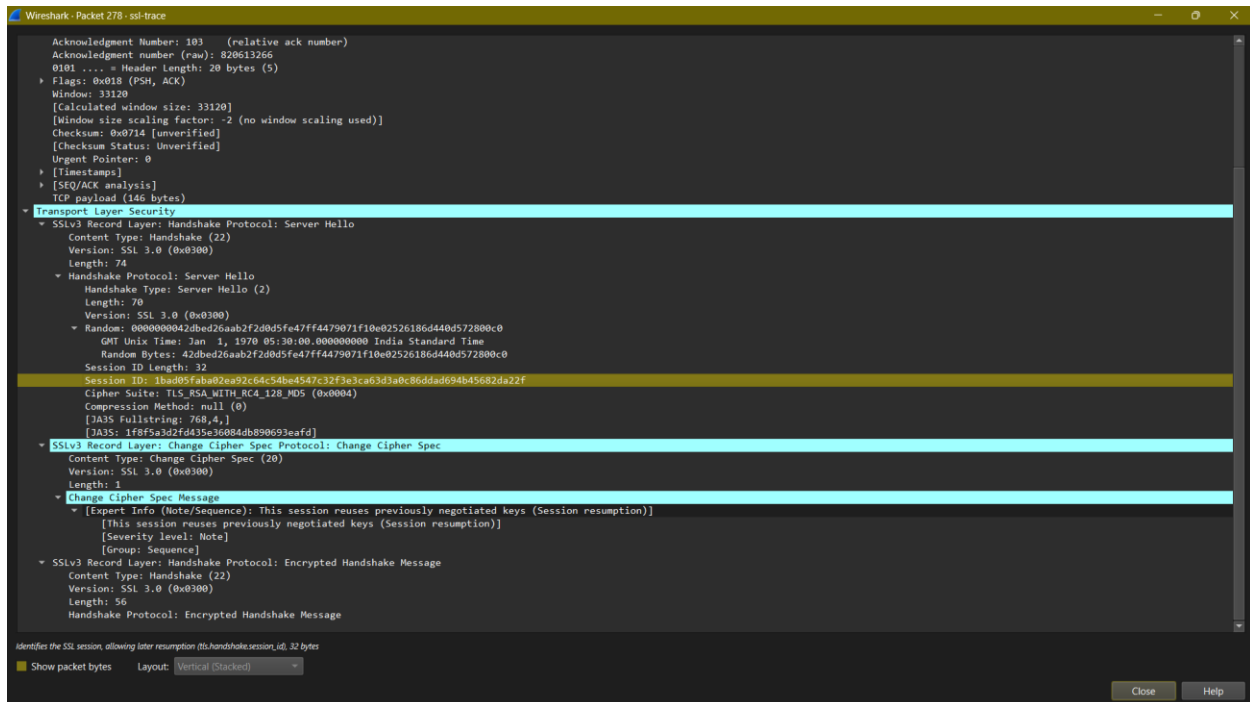
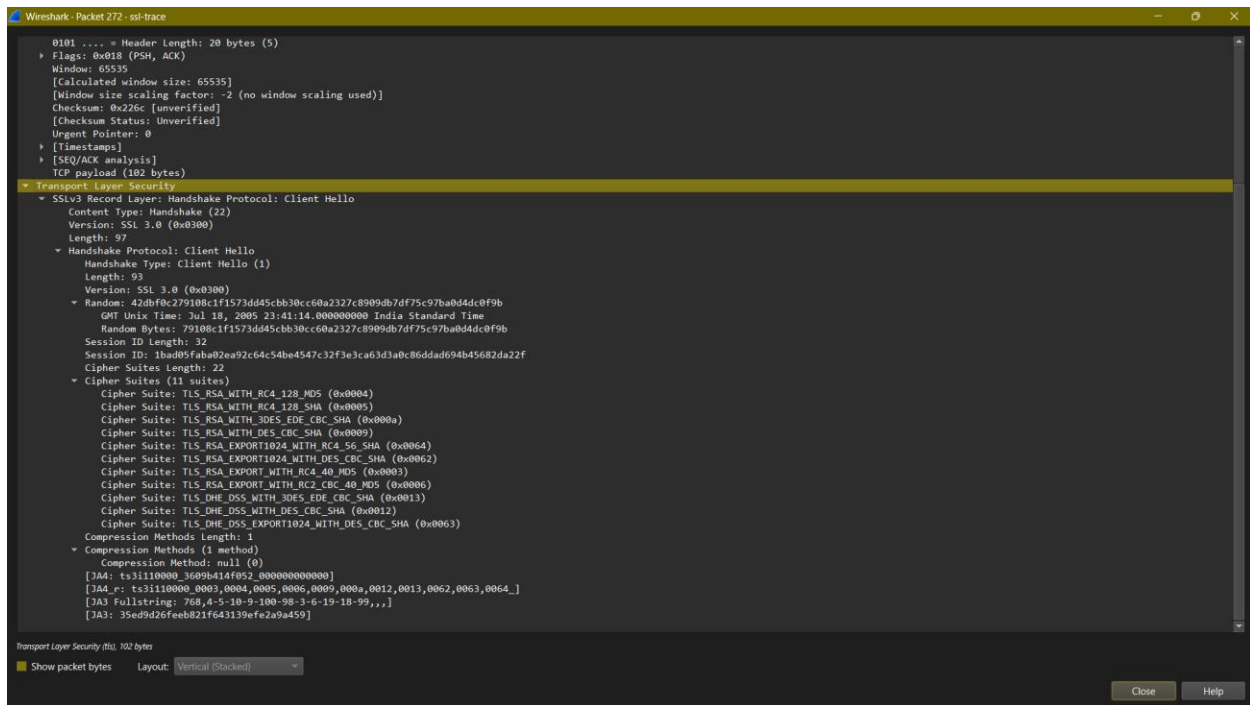
A client (web browser) and a server (web server) must create a secure communication session using the SSL handshake. The client sends a "hello" message to the server with the SSL/TLS version, available cypher suites, and a "client random." With its "hello" message, the server specifies its SSL/TLS version, cypher suite, SSL certificate, and "server random" number. The client receives the server's SSL certificate for authentication. The client and server then exchange keys using RSA or Diffie-Hellman to create a shared secret. This shared secret generates session keys for data encryption during the session. The client and server send "finished" messages to conclude the handshake and start the secure session. This procedure encrypts and secures client-server connection, preventing data eavesdropping and tampering.

### 2.2 In-Depth Comparative Evaluation of SSL Trace

Message Type	SSL Version	Frame Number	Source	Destination	Details
Client Hello	SSLv2	106	128.23 8.38.16 2	216.75. 194.22 0	Version: SSL 2.0 Random: Randomly generated number Cipher Suites: SSL2_RC4_128_WITH_MD5 SSL2_DES_192_EDE3_CBC_WITH_MD5 SSL2_RC2_128_CBC_WITH_MD5 SSL2_DES_64_CBC_WITH_MD5 SSL2_RC4_128_EXPORT40_WITH_MD5 SSL2_RC2_128_CBC_EXPORT40_WITH_MD5

Server Hello	SSLv3	108	216.75.194.220	128.238.38.162	Version: SSL 3.0 Random: Randomly generated number Session ID: Not explicitly mentioned in the trace. Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 Compression Method: null
Client Hello	SSLv3	272	128.238.38.162	216.75.194.220	Version: SSL 3.0 Random: Randomly generated number Cipher Suites: TLS_RSA_WITH_RC4_128_MD5 TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_3DES_EDE_CBC_SHA TLS_RSA_WITH_DES_CBC_SHA TLS_RSA_EXPORT1024_WITH_RC4_56_SHA TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA TLS_RSA_EXPORT_WITH_RC4_40_MD5 TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA TLS_DHE_DSS_WITH_DES_CBC_SHA TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA
Server Hello	SSLv3	278	216.75.194.220	128.238.38.162	Version: SSL 3.0 Random: 0000000042dbed26aab2f22d0d5fe47 SessionID:1bad05faba02ea92c64c54be4547c32f3e3ca63d3a0c86ddad694b45682da22f Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 Compression Method: null





## 2.3 In-Depth Security Aspects

### 2.3.1 Chosen Cipher Suites

The SSL trace file indicates that the negotiated cipher suite for the session is TLS\_RSA\_WITH\_RC4\_128\_MD5 (0x0004) for SSL 3.0. This is the sole cypher suite used in the session, as indicated by the "Server Hello" message. The trace for SSL 2.0 does not indicate a distinct negotiated cipher suite. The same cipher suite, TLS\_RSA\_WITH\_RC4\_128\_MD5, is employed consistently during the session.

#### **"TLS\_RSA\_WITH\_RC4\_128\_MD5"**

Component	Details
Key Exchange	RSA
Key Length	Typically, 2048 bits or higher
Security	RSA is widely used and considered secure for key exchange, but it does not provide forward secrecy
Encryption	RC4 (128-bit)
Key Length	128 bits
Encryption Strength	RC4 is a stream cipher known for its speed and simplicity. However, it has several vulnerabilities, including the RC4 bias attack, which makes it less secure compared to modern ciphers
Resistance to Attacks	RC4 is vulnerable to several attacks, such as the RC4 bias attack and the BEAST attack. It is generally not recommended for use in new systems
Hashing	MD5
Key Length	128 bits
Security	MD5 is considered weak due to its susceptibility to collision attacks. It is not recommended for use in cryptographic applications where data integrity is critical

### 2.3.2 Security Strength Analysis

The server chose this cipher suite as it was among the options provided by the client in the "Client Hello" message. The server must select a cipher suite that is mutually supported by both the client and server to establish a secure connection. The RC4 algorithm is known for its speed and efficiency, providing it an ideal selection for scenarios where performance is paramount. Despite its known weaknesses, RC4 was extensively used throughout this period due to its performance advantages. At the time of this trace, TLS\_RSA\_WITH\_RC4\_128\_MD5 was considered as a secure choice. It employs RSA for key exchange, RC4 for encryption, and MD5 for message integrity verification. Although MD5 and RC4 are presently regarded as vulnerable, they were widely used in earlier times. The trace suggests that the session used previously negotiated keys, indicating that the server may have selected this cipher suite to ensure consistency with a prior session, thus enhancing efficiency by avoiding a complete handshake.

### **2.3.3 Justification for Choosing TLS\_RSA\_WITH\_RC4\_128\_MD5 Cipher Suite**

TLS\_RSA\_WITH\_RC4\_128\_MD5 was extensively used and deemed secure in the early 2000s. It offered an optimal equilibrium of performance and security for numerous applications. Over time, weaknesses were identified in both RC4 and MD5. RC4 was discovered to be vulnerable to specific attack vectors that might compromise the secrecy of the encrypted information. MD5 has been identified as susceptible to collision attacks, where two different inputs might produce identical hash values, hence compromising its efficacy in guaranteeing data integrity. The SSL trace packets were recorded on July 18, 2005, starting at 23:41:12 India Standard Time (IST), equivalent to 18:11:12 UTC. The vulnerabilities in the RC4 cypher were identified gradually, with major vulnerabilities observed around 2013. Investigations indicated that RC4 was vulnerable to multiple attack vectors, resulting in being outdated with modern security protocols. The deficiencies in the MD5 hashing technique had been identified previously, with notable problems detected as early as 2004. Collision attacks, in which two distinct inputs yield identical hash values, were a significant problem.

### **2.3.4 Modern Security Practices**

Modern security protocols advise against the use of RC4 and MD5 because of their established vulnerabilities. Rather, more robust techniques such as AES (Advanced Encryption Standard) for encryption and SHA-256 (Secure Hash Algorithm 256-bit) for hashing are preferred. Using more secure cipher suites, such as those founded on AES and SHA-256, mitigates present threats and guarantees a higher degree of security. In conclusion, although TLS\_RSA\_WITH\_RC4\_128\_MD5 was deemed secure during the trace, progress in cryptographic research and the identification of flaws have resulted in the implementation of more resilient encryption techniques in modern security protocols.

## Section 3: Kerberos

### **3.1 Kerberos**

Kerberos is a network authentication protocol that guarantees safe authentication for client/server applications via secret-key cryptography. It comprises three essential components: the Key Distribution Centre (KDC), which comprises the Authentication Server (AS) and the Ticket Granting Server (TGS); the client, representing the user or application seeking access; and the server, which delivers the requested service. The authentication procedure begins with the client transmitting its credentials to the AS, which authenticates them and issues a Ticket Granting Ticket (TGT). The client subsequently uses the TGT to request access to a designated service from the TGS, which authenticates the TGT and provides a service ticket. The client ultimately submits this service ticket to the server, which decrypts it and provides access if it is valid. Kerberos strengthens security by preventing password transmission over the network, facilitates Single Sign-On (SSO) for effortless access to various services, and offers mutual authentication, enabling both the client and server to validate each other's identities.

### **3.2 In-Depth Analysis of Kerberos Traces**

#### **3.2.1 AS-REQ (Authentication Service Request)**

The AS-REQ (Authentication Service Request) message is the preliminary request submitted by the client to the Authentication Server (AS) to acquire a Ticket Granting Ticket (TGT). The AS-REQ message contents from the file includes:

**Source IP:** 10.1.12.2

**Destination IP:** 10.5.3.1



**Protocol Version Number (pvno):** 5

**Message Type (msg-type):** krb-as-req (10)

**Pre-Authentication Data (padata):** Contains two items:

**PA-ENC-TIMESTAMP:** An encrypted timestamp that use the client's secret key.

**PA-PAC-REQUEST:** Specifies whether the client requests a Privilege Attribute Certificate (PAC).

**Request Body (req-body):** Includes various options and parameters such as:

**KDC Options (kdc-options):** 40810010 (forwardable, renewable, canonicalize)

**Client Name (cname):** des

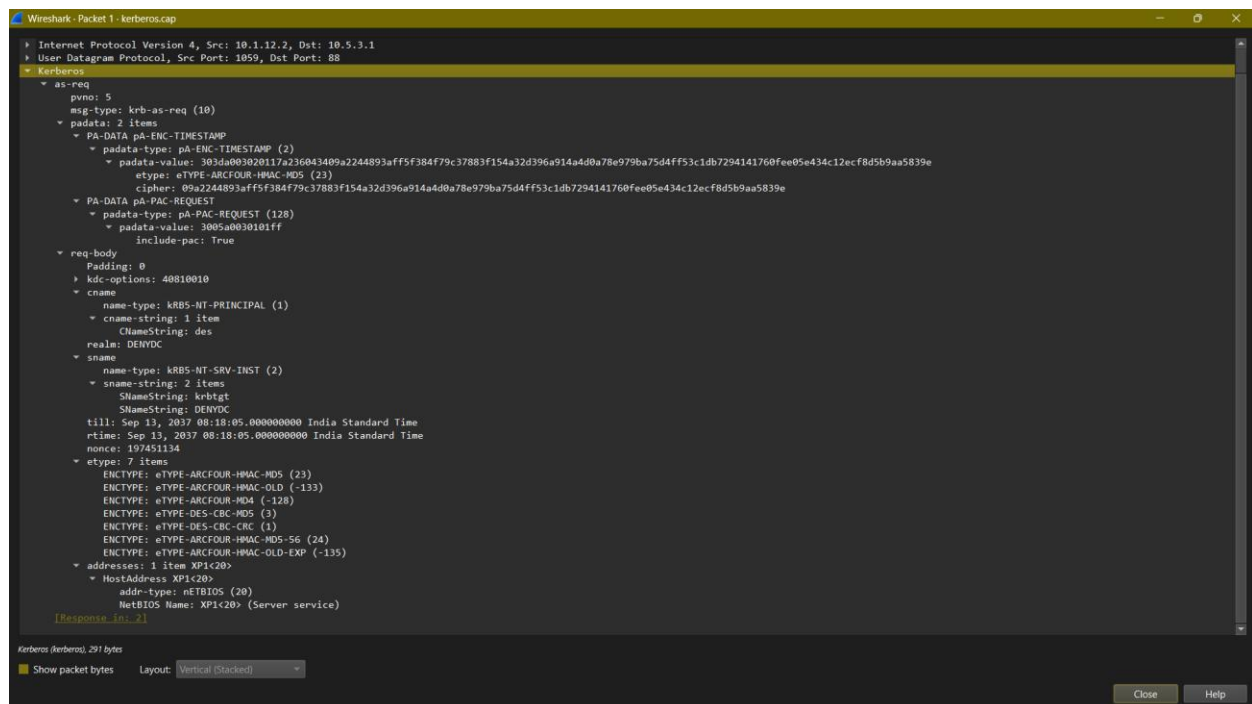
**Realm (realm):** DENYDC

**Service Name (sname):** krbtgt

**Till:** Sep 13, 2037 08:18:05.000000000 India Standard Time

**Nonce:** 197451134

**Encryption Types (etype):** eTYPE-ARCFOUR-HMAC-MD5 (23), eTYPE-ARCFOUR-HMAC-OLD (-133), eTYPE-ARCFOUR-MD4 (-128), eTYPE-DES-CBC-MD5 (3), eTYPE-DES-CBC-CRC (1), eTYPE-ARCFOUR-HMAC-MD5-56 (24), eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)



### 3.2.2 AS-REP (Authentication Service Reply)

The AS-REP (Authentication Service Reply) message is a response from the Authentication Server (AS) to the client's AS-REQ (Authentication Service Request). The AS-REP message contents from the file include:

**Source IP:** 10.5.3.1

**Destination IP:** 10.1.12.2

**Protocol Version Number (pvno):** 5

**Message Type (msg-type):** krb-as-rep (11)

**Pre-Authentication Data (padata):** Contains the password salt.

**Client Realm (crealm):** DENYDC.COM

**Client Name (cname):** des

**Ticket (ticket):** The Ticket Granting Ticket (TGT), which includes:

**Ticket Version Number (tgt-vno):** 5

**Realm (realm):** DENYDC.COM

**Service Name (sname):** krbtgt

**Encrypted Part (enc-part):** eTYPE-ARCFOUR-HMAC-MD5 (23), kvno: 2,  
cipher: [encrypted data]

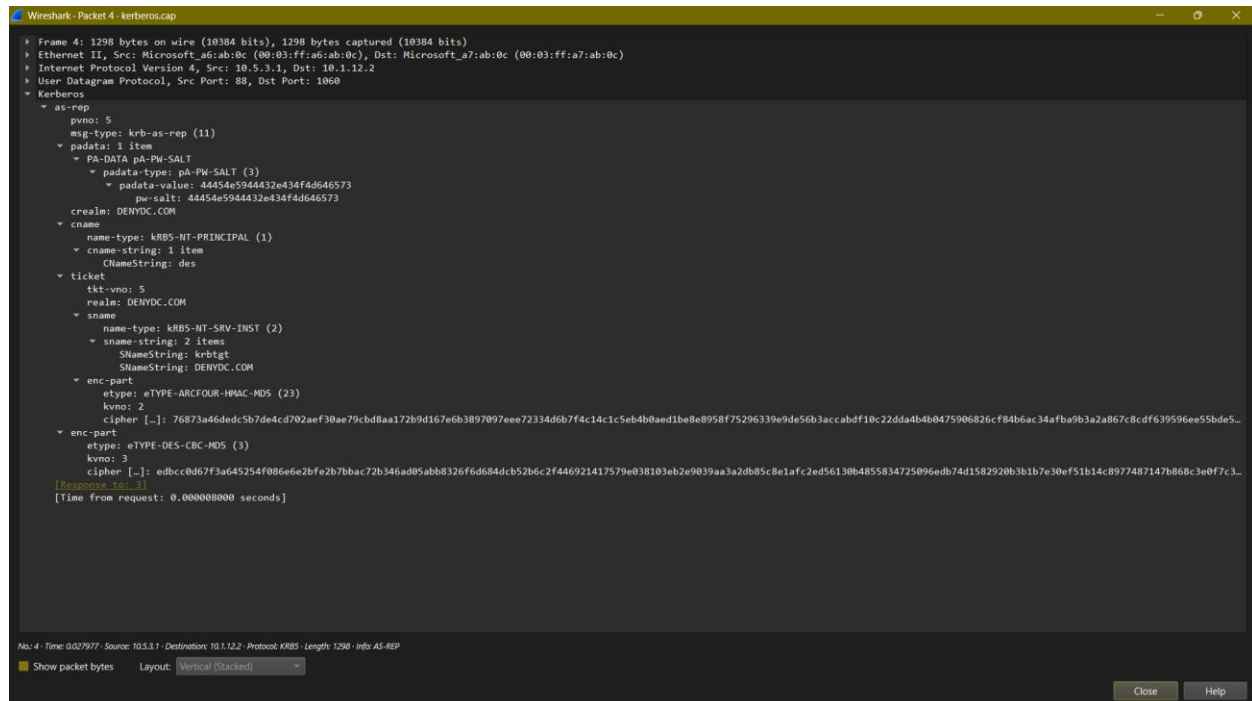
**Encrypted Part (enc-part):** Encrypted with the client's secret key, it includes:

**Session Key:** A key shared between the client and the TGS.

**Ticket Flags:** Flags indicating various options like forwardable, renewable, etc.

**Client Address:** The client's address.

**Ticket Lifetime:** The validity period of the ticket.



### 3.2.3 TGS-REQ (Ticket Granting Service Request)

**Encryption Types (etype):** eTYPE-ARCFOUR-HMAC-MD5 (23), eTYPE-ARCFOUR-HMAC-OLD (-133), eTYPE-ARCFOUR-MD4 (-128), eTYPE-DES-CBC-MD5 (3), eTYPE-DES-CBC-CRC (1), eTYPE-ARCFOUR-HMAC-MD5-56 (24), eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)



The TGS-REP (Ticket Granting Service Reply) communication constitutes the response from the TGS to the client's TGS-REQ. The TGS-REP message contents from the file include:

**Source IP:** 10.5.3.1

**Destination IP:** 10.1.12.2

**Protocol Version Number (pvno):** 5

**Message Type (msg-type):** krb-tgs-rep (13)

**Client Realm (crealm):** DENYDC.COM

**Client Name (cname):** des

**Ticket (ticket):** The service ticket, which includes:

**Ticket Version Number (tgt-vno):** 5

**Realm (realm):** DENYDC.COM

**Service Name (sname):** host/xp1.denydc.com

**Encrypted Part (enc-part):** eTYPE-ARCFOUR-HMAC-MD5 (23), kvno: 2, cipher: [encrypted data]

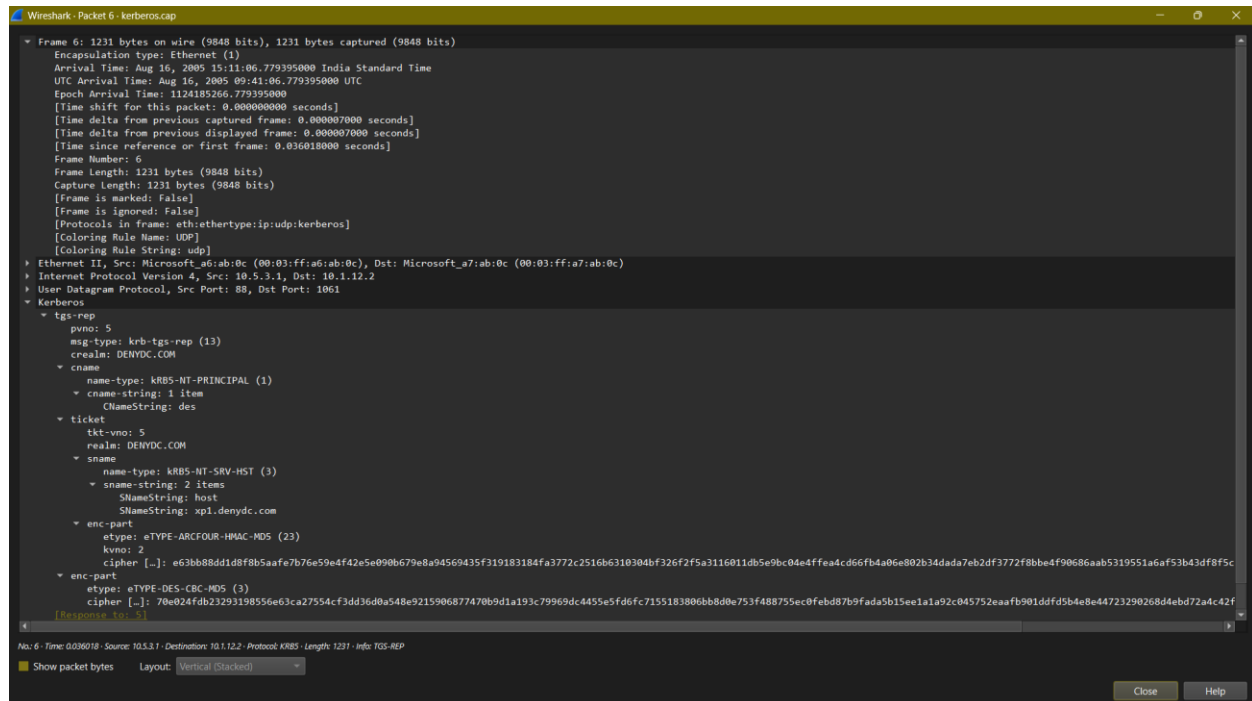
**Encrypted Part (enc-part):** Encrypted with the client's session key, it includes:

**Session Key:** A key shared between the client and the service server.

**Ticket Flags:** Flags indicating various options like forwardable, renewable, etc.

**Client Address:** The client's address.

**Ticket Lifetime:** The validity period of the ticket.



### 3.3 In-Depth Security Features

### 3.3.1 Algorithms

Kerberos uses many cryptographic techniques to guarantee secure authentication and communication.

**AES (Advanced Encryption Standard):** Employed for the encryption of tickets and session keys. AES is known for its robust security and efficiency.

**HMAC (Hash-based Message Authentication Code):** Employed for integrity verification. HMAC guarantees the integrity of the data during transmission.

**DES (Data Encryption Standard):** Despite being antiquated and less secure, DES was employed in early iterations of Kerberos. It has predominantly been supplanted by AES owing to its flaws.

### 3.3.2 Security Strength

Kerberos offers comprehensive security via multiple mechanisms:

Kerberos employs timestamps to mitigate replay attacks. Every ticket contains a timestamp and an expiration duration, preventing the reuse of expired tickets.

**Time-Limited Tickets:** In Kerberos, tickets possess a restricted validity duration, generally ranging from 8 to 10 hours. This temporal strategy guarantees that a compromised ticket cannot be used perpetually.

**Mutual Authentication:** Both the client and server verify each other's identities, hence diminishing the likelihood of man-in-the-middle attacks.

**Session Keys:** Kerberos produces distinct session keys for every session, guaranteeing that the compromise of a single session key does not impact other sessions.

Kerberos is a robust authentication technology, extensively employed in numerous applications to safeguard sensitive data and facilitate secure communication.

### 3.3.3 Kerberos Limitations

#### Single point of failure

The Key Distribution Centre (KDC) retains the secret keys for all users and services; thus, if the KDC is compromised, attackers can obtain user credentials.

## **Password guessing**

Attackers can request many tickets to gather information similar to `/etc/passwd`, as no authentication is required to request a ticket

## **Cross-realm authentication**

Kerberos inadequately facilitates cross-realm authentication; thus, customers are required to procure a ticket from each realm's Key Distribution Centre to access services across different realms.

## **Group limit**

Kerberos is restricted to 1015 groups, including nested groups. An error occurs when the user logs into Windows if the number of groups is exceeded.

# **Section 4: DNS Anomalies**

## **4.1 DNS Protocol**

The Domain Name System (DNS) converts human-readable domain names (e.g., `www.example.com`) into IP addresses (e.g., `192.0.2.1`) used by computers for network identification. Upon entering a URL into your browser, a DNS query is executed to find the relevant IP address. This inquiry initially proceeds to a DNS resolver (recursive DNS server), which functions as an intermediary between the client and the DNS servers. If the resolver lacks the response, it requests a root DNS server, which directs it to the relevant top-level domain (TLD) server (e.g., `.com`, `.org`). The TLD server then instructs the resolver to the authoritative DNS server for the domain. The authoritative DNS server supplies the IP address for the requested domain, and the resolver relays the IP address to the client, enabling the browser to connect to the web server. Various categories of DNS servers exist: Recursive DNS servers process client queries and execute the requisite lookups to resolve domain names; authoritative DNS servers furnish responses to queries concerning domains for which they hold responsibility; root DNS servers serve as the initial stage in converting human-readable domain names into IP addresses; and TLD DNS servers oversee top-level domains and route queries to the corresponding authoritative servers.

## **4.2 In-Depth Analysis of DNS Trace 1**

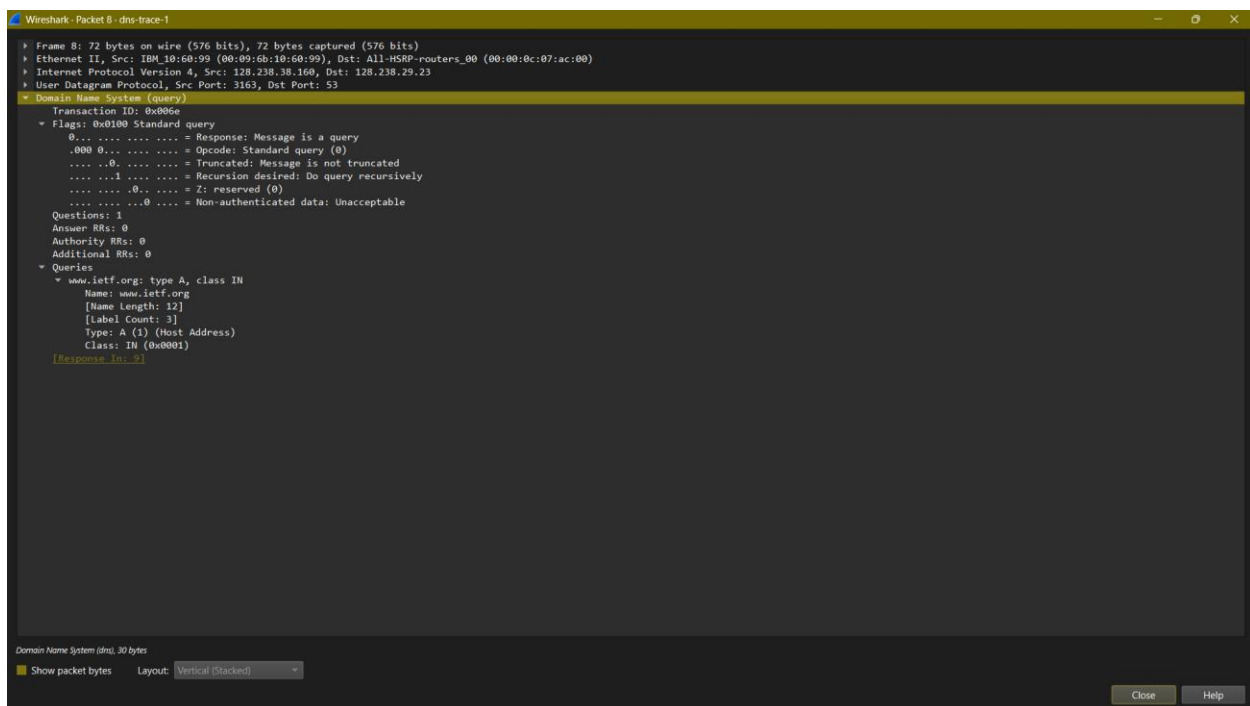
### 4.2.1 DNS Request

**Source IP:** 128.238.38.160

**Destination IP:** 128.238.29.23

**Query:** Standard query 0x006e A [www.ietf.org](http://www.ietf.org)

This DNS query requests the IP address linked to the domain [www.ietf.org](http://www.ietf.org). The query type is an A record, used for obtaining the IP address of a host.



### 4.2.2 DNS Response

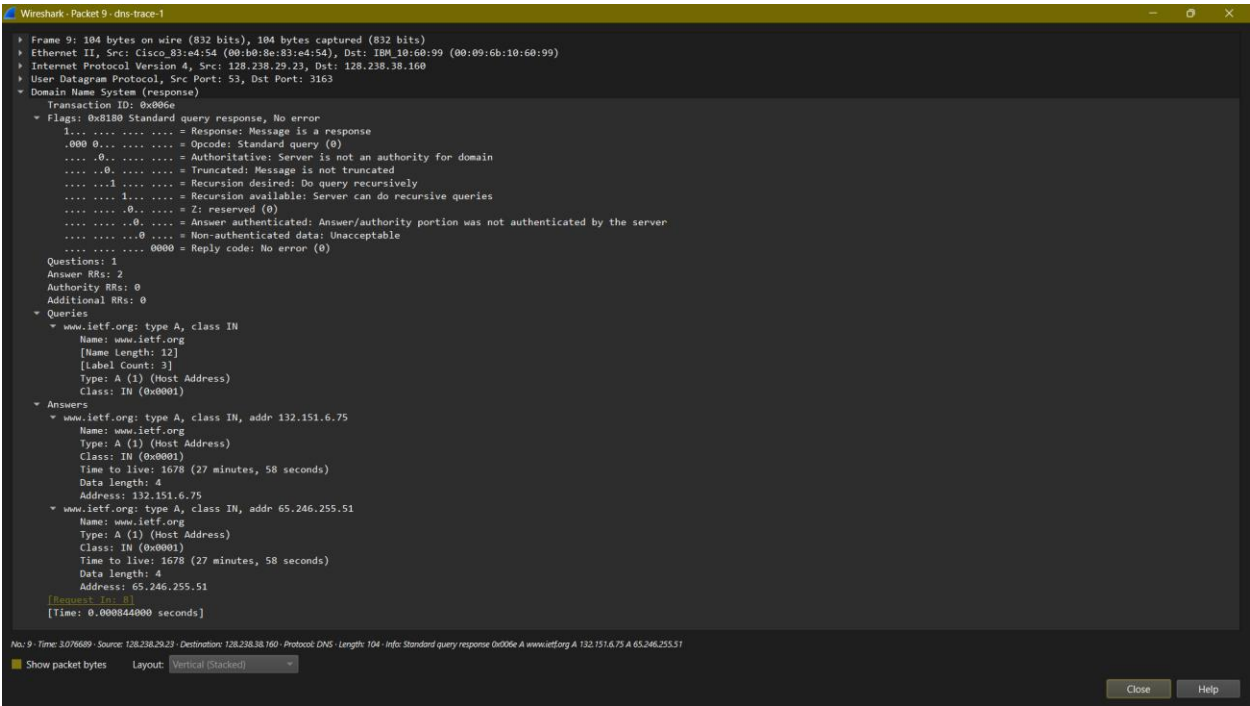
**Source IP:** 128.238.29.23

**Destination IP:** 128.238.38.160

**Response:** Standard query response 0x006e A [www.ietf.org](http://www.ietf.org) A 132.151.6.75 A 65.246.255.51

The DNS answer includes the IP addresses for the specified domain. In this instance, [www.ietf.org](http://www.ietf.org) corresponds to two IP addresses: 132.151.6.75 and 65.246.255.51. This

signifies that the domain has many A records, which may be used for load balancing or redundancy.



4.3 Comparative Evaluation of Normal DNS and Anomalous DNS Pair

Category	Field	Normal DNS Pair	Anomalous DNS Pair
Request	Source IP	192.168.1.3	192.168.1.3
	Destination IP	192.168.1.1	192.168.1.1
	Source Port	1393	1394
	Destination Port	53 (DNS)	53 (DNS)
	Transaction ID	0x0001	0x0002
	Query	PTR record for 1.1.168.192.in-addr.arpa	A record for <a href="#">www.ietf.org</a>
Response	Source IP	192.168.1.1	192.168.1.1
	Destination IP	192.168.1.3	192.168.1.3
	Source Port	53 (DNS)	53 (DNS)



	Destination Port	1393	1394
	Transaction ID	0x0001	0x0002
	Response	PTR record for 1.1.168.192.in-addr.arpa resolves to SpeedTouch.lan	No error, but domain <a href="http://www.www.com.lan">www.www.com.lan</a> is unusual

## 4.4 Normal DNS Pair

### 4.4.1 Query

Wireshark - Packet 2 - dns-anomaly.pcap

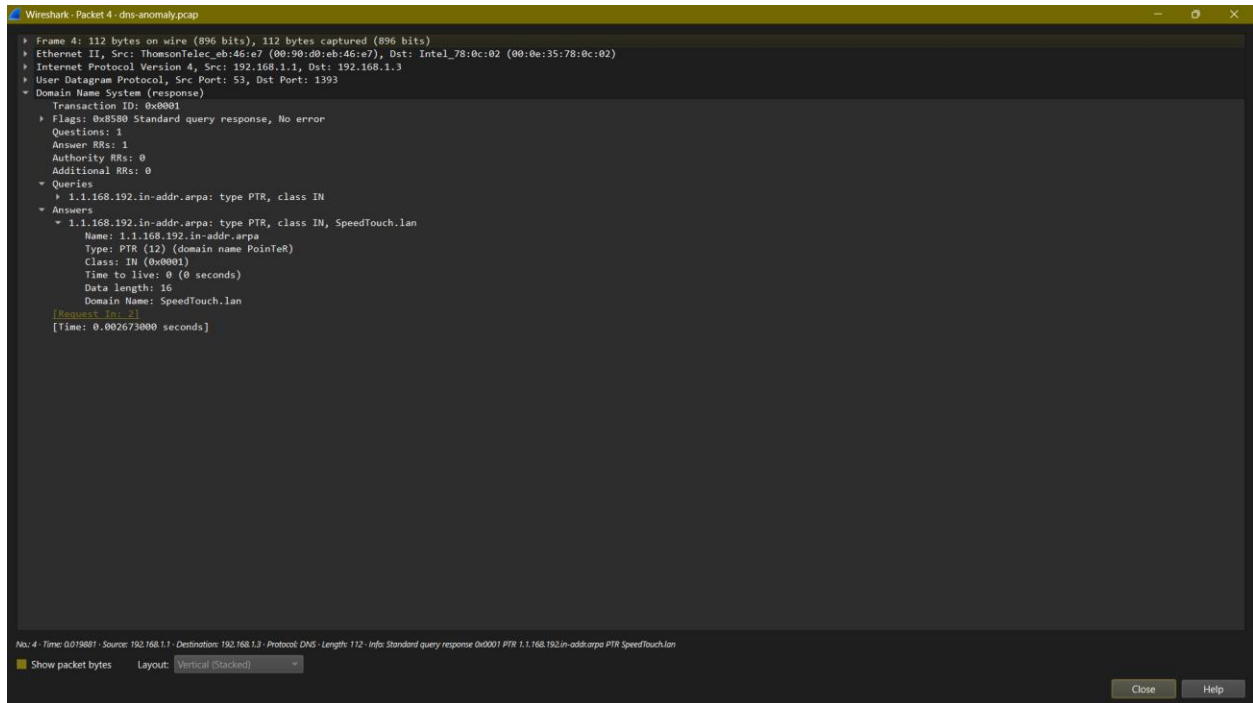
```

Frame 2: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface 0
Ethernet II, Src: Intel_78:0c:02 (00:0e:35:78:0c:02), Dst: ThomsonTelec_eb:46:e7 (00:90:d8:eb:46:e7)
Internet Protocol Version 4, Src: 192.168.1.3, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 1393, Dst Port: 53
Domain Name System (query)
  Transaction ID: 0x0001
  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
  Queries
    1.1.168.192.in-addr.arpa: type PTR, class IN
      Name: 1.1.168.192.in-addr.arpa
      [Name Length: 24]
      [Label Count: 6]
      Type: PTR (12) (domain name PoinTeK)
      Class: IN (0x0001)
    [Response: 192.168.1.1]
  
```

Packet 2 - Time: 0.017208 - Source: 192.168.1.3 - Destination: 192.168.1.1 - Protocol: DNS - Length: 84 - Info: Standard query 0x0001 PTR 1.1.168.192.in-addr.arpa

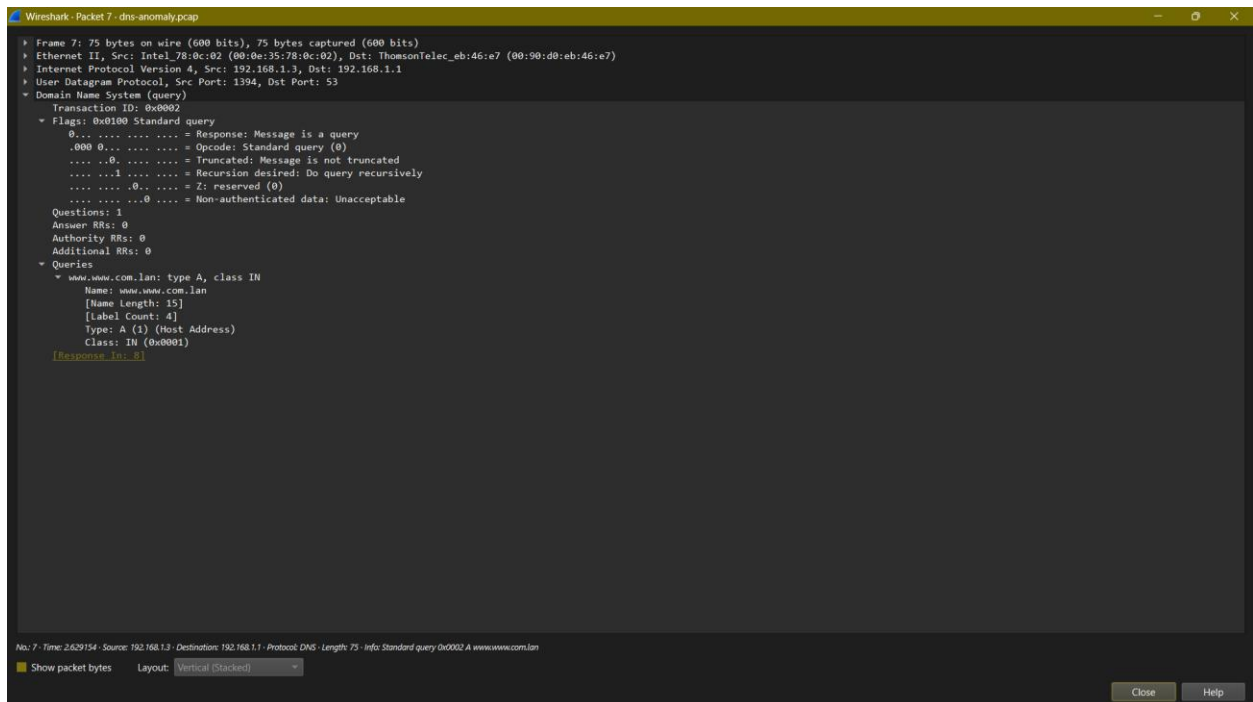
Show packet bytes    Layout: Vertical (Stacked)    Close    Help

### 4.4.2 Response

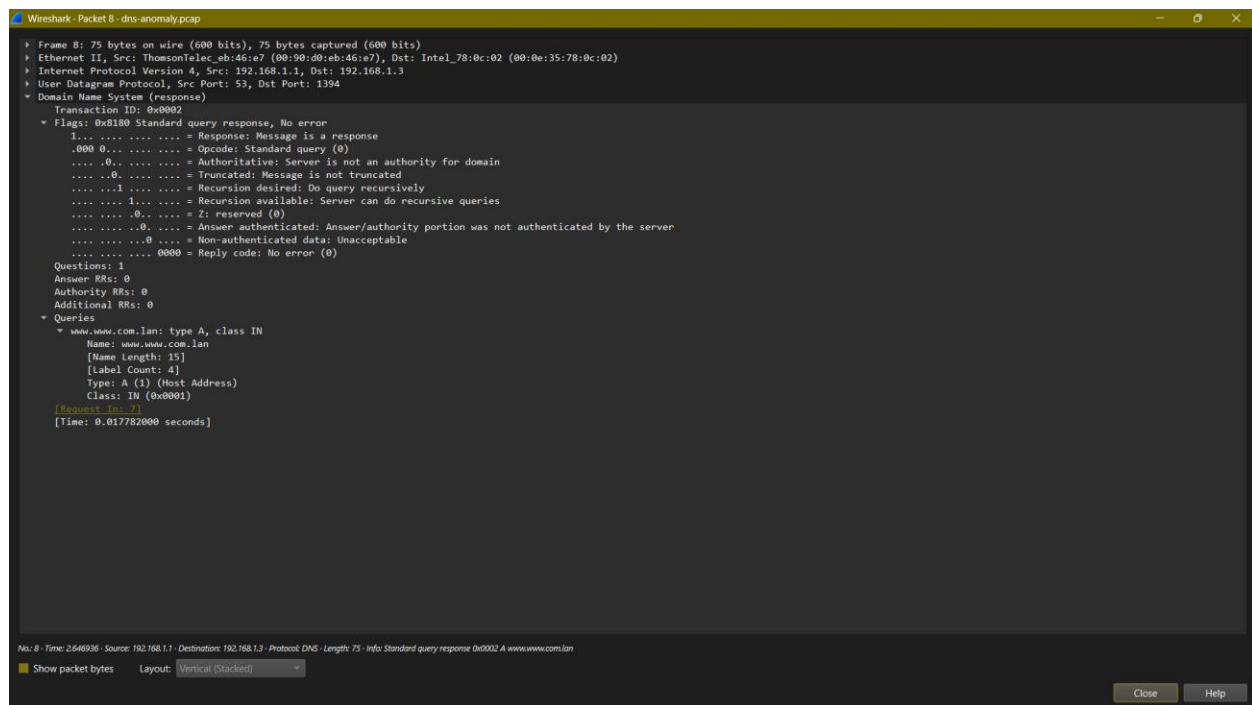


## 4.5 Anomalous DNS Pair

### 4.5.1 Query



### 4.5.2 Response

A screenshot of the Wireshark network protocol analyzer interface. The title bar reads 'Wireshark - Packet 8 - dns-anomaly.pcap'. The main display area shows the details of a DNS response packet (Packet 8). The packet structure is as follows:

- Frame 8: 75 bytes on wire (600 bits), 75 bytes captured (600 bits)
- Ethernet II, Src: ThomsonTelec\_eb46:e7 (00:90:db:eb46:e7), Dst: Intel\_78:0c:02 (00:0e:35:78:0c:02)
- Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.3
- User Datagram Protocol, Src Port: 53, Dst Port: 1394
- Domain Name System (response)
  - Transaction ID: 0x0002
  - Flags: 0xB180 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: 0
  - Authority RRs: 0
  - Additional RRs: 0
- Queries
  - 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)

The status bar at the bottom shows: 'No. 8 - Time: 2.646936 - Source: 192.168.1.1 - Destination: 192.168.1.3 - Protocol: DNS - Length: 75 - Info: Standard query response 0x0002 A www.www.com.lan'. The 'Show packet bytes' button is active, and the layout is set to 'Vertical (Stacked)'. 'Close' and 'Help' buttons are in the bottom right corner.

## 4.6 Nature and Detection of Anomaly

The query regarding `www.www.com.lan` is unusual and does not conform to common domain naming conventions. This may suggest a misconfiguration or an effort to address a nonexistent domain. The irregularity is found by the examination of the domain name in the query. The recurrence of 'www' and the '.lan' top-level domain shows that this is not a standard DNS query. The quick repetition of analogous queries (e.g., `www.www.com`) may signify suspicious behavior.

## 4.7 DNS security limitations

**Absence of Encryption:** Conventional DNS queries and responses are transmitted in plaintext, making them susceptible to interception and eavesdropping by malicious entities.

**DNS Spoofing/Cache Poisoning:** Attackers can inject counterfeit DNS information into a resolver's cache, resulting in the return of incorrect IP addresses. This may redirect people to harmful websites.

**DNS Hijacking:** This entails the redirection of DNS queries to an alternative domain name server, either by malware or through unauthorised alterations of a DNS server.

**NXDOMAIN Attack:** This category of DNS flood attack overwhelms a DNS server with requests for non-existent records, potentially resulting in a denial-of-service for real traffic.

## 5. References

1. Wikipedia. (n.d.). Transmission Control Protocol. [online] Available at: [https://en.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) [Accessed 2 Jan. 2025].
2. Wikipedia. (n.d.). Transport Layer Security. [online] Available at: [https://en.wikipedia.org/wiki/Transport\\_Layer\\_Security](https://en.wikipedia.org/wiki/Transport_Layer_Security) [Accessed 2 Jan. 2025].
3. Wikipedia. (n.d.). Kerberos (protocol). [online] Available at: [https://en.wikipedia.org/wiki/Kerberos\\_\(protocol\)](https://en.wikipedia.org/wiki/Kerberos_(protocol)) [Accessed 2 Jan. 2025].
4. Wikipedia. (n.d.). Domain Name System. [online] Available at: [https://en.wikipedia.org/wiki/Domain\\_Name\\_System](https://en.wikipedia.org/wiki/Domain_Name_System) [Accessed 12 Jan. 2025].
5. IETF. (2023). TLS Subcerts. [online] Available at: <https://datatracker.ietf.org/doc/html/draft-ietf-tls-subcerts-15> [Accessed 2 Jan. 2025].
6. Scholarship. (n.d.). [online] Available at: <https://jscholarship.library.jhu.edu/items/00493bcd-8e2e-4ac5-8803-7022a0275b3f> [Accessed 5 Jan. 2025].
7. Cloudfront. (2009). Designed for Change. [pdf] Available at: <https://d1bcsfjk95uj19.cloudfront.net/files/2009-designed-for-change.pdf> [Accessed 5 Jan. 2025].
8. IETF. (1981). Internet Protocol. [online] Available at: <https://datatracker.ietf.org/doc/html/rfc791> [Accessed 13 Jan. 2025].
9. MIT. (n.d.). Kerberos: The Network Authentication Protocol. [online] Available at: <https://web.mit.edu/kerberos/krb5-1.21/> [Accessed 6 Jan. 2025].
10. IONOS. (n.d.). Kerberos. [online] Available at: <https://www.ionos.com/digitalguide/server/security/kerberos/> [Accessed 7 Jan. 2025].
11. Google Books. (n.d.). [online] Available at: [https://books.google.co.in/books?id=5PJisOKJ0k8C&pg=PA22&redir\\_esc=y#v=onepage&q&f=false](https://books.google.co.in/books?id=5PJisOKJ0k8C&pg=PA22&redir_esc=y#v=onepage&q&f=false) [Accessed 13 Jan. 2025].
12. Wiley Online Library. (2014). [online] Available at: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2014.0386> [Accessed 7 Jan. 2025].