Suspicious Traffic Detection

By - Ahmed Medhat

Main Point

Here, we are going to discuss different types of traffic that are used frequently in network communication and could be misused by malicious actors to achieve their goals.

When analyzing network traffics or analyzing network forensics, especially when using Wireshark, we must focus our understanding on the core types of protocols, their structure form and how do they work, so that we can distinguish between their natural form or misuse of them.

So, the main idea here is the understanding the core network protocols that used frequently at any connection.

There is no doubt that there are other harmful messages traveling across the network, but our focus here is on the type of traffic and not the content of the message itself.

1- DHCP

Structure:

Offsets	Octet	0	1	2	3
Octet	Bit	0–7	8–15	16–23	24–31
0	0	OpCode	Hardware Type	Hardware Length	Hops
4	32	Transaction ID			
8	64	Seconds Elapsed		Flags	
12	96	Client IP Address			
16	128	Your IP Address			
20	160	Server IP Address			
24	192	Gateway IP Address			
28	224	Client IP Address			
32	256	Client Hardware Address (16 bytes)			
36	288				
40	320				
44	352				
48+	384+	Server Host Name (64 bytes)			
		Boot File (128 bytes)			
		Options			

OPCODE: Tells us the type of the packet if it is request or reply
Hardware type: the type of hardware address, Ex: Ethernet, IEEE802, ATM
Hardware length: the length of the hardware address
Hops: Hops or routers which help in finding a DHCP server
Transaction ID: Random number, it is unique for request and its response
Seconds Elapsed: no of Seconds started when the client first requested an address from the DHCP server
Flags: The type of traffic the DHCP client can accept. (unicast, broadcast)
Client IP Address: The client's new IP address from Offer message.
Your IP Address: The IP address offered by the DHCP server
Server IP Address: The DHCP server's IP address
Gateway IP Address: The IP address of the network's default gateway
Client Hardware Address: The client's MAC address

Server Host Name: (optional) The server's host name

Boot File: (optional) A boot file for use by DHCP

Options: Used to expand the structure of the DHCP with more features.

1: Subnet mask	15: Domain name
3: Router	31: Router discovery
4: Time server	50: Requested IP address
5: Name server	53: DHCP message type
6: Domain server	54: DHCP server identifier (IP)
12: Host name	255: end

Normal:

- It's referenced to BOOTP protocol which is client/server protocol
- Transaction id is unique for the request and its response
- Notice the natural sequence of DHCP messages (D,O,R,A)
- For Discover message,
  - Source and client IP is 0.0.0.0
  - Destination IP is 255.255.255.255 (broadcast)
- For Offer and Ack messages,
  - Source IP is (DHCP server IP)
  - Client IP is (New IP obtained)
  - Destination IP is 255.255.255.255 (broadcast)
- For Request message,
  - Source IP is 0.0.0.0
  - Client IP is (New IP obtained)
  - Destination IP is 255.255.255.255 (broadcast)
- Uses UDP on port 67 if server, 68 if client

Suspicious: Huge number of DHCP requests from different and suspicious IP addresses (DHCP starvation) or spoofed DHCP IP address.

2- ARP

Structure:

Hardware type (2 bytes)		Protocol type (2 bytes)
Hardware address length (1 byte)	Protocol address length (1 byte)	Operation code (2 bytes)
Source hardware address*		
Source protocol address*		
Target hardware address*		
Target protocol address*		

Hardware type: the type of hardware address, Ex: Ethernet, IEEE802, ATM
Protocol type: the type of protocol (IPv4 or IPv6)
Hardware length: the length of the hardware address
Protocol length: the length of the IP address
Operation code: (1) for ARP request, (2) for ARP reply
Source hardware address: sender mac address
Source protocol address: sender IP address
Target hardware address: receiver mac address
Target protocol address: receiver IP address

Normal:

- A normal ARP Request typically follows a response
- Target mac address is FF:FF:FF:FF:FF:FF in the ARP request message

Suspicious:

- Huge number of ARP requests in small time (ARP sweep)
- ARP requests come from suspicious IPs that are not seem to be real. Ex: 192.168.1.3, 192.168.1.**4**, 192.168.1.**5**, 192.168.1.**6**, … (Notice that the number is increased by one, this is not usual behavior)
- Notice Mac spoofing which means there are 2 identical mac address with different Ips.
- ARP replay without ARP request (ARP spoofing using gratuitous ARP reply)

3- ICMP

Structure:

Bit 0–7	Bit 8–15	Bit 16–23	Bit 24–31
Type	Code	Checksum	
Header Information			

Type: Tells us the description of the message

0: Echo reply.
3: Destination unreachable.
5: Redirect Message.
8: Echo Request.
11: Time Exceeded.
12: Parameter problem.

Code: tells us information about error message and type

Checksum: message integrity check, it detects whether any error exists in the message or not.

Header Information: Information about the message depend on reply type

Normal:

- Echo request has **code: 0** and **Type: 8**
- Notice **Data** section which has a random value, this random value is unique for request and its reply

Suspicious:

- The Data section can be used as exfiltration channel.
- The number of sent packets. (ping flood) (DOS)
- Notice any ICMP response from external network you did not make as it could be part of reconnaissance plan.

4- TCP

Structure:

<table>
<tr><td colspan="8">Source port (16 bits)</td><td colspan="8">Destination port (16 bits)</td></tr>
<tr><td colspan="16">Sequence number (32 bits)</td></tr>
<tr><td colspan="16">Acknowledge number (32 bits)</td></tr>
<tr><td colspan="4">Header Length (4 bits)</td><td colspan="3">Reserved bits (4 bits)</td><td>U R G</td><td>A C K</td><td>P S H</td><td>R S T</td><td>S Y N</td><td>F Y N</td><td colspan="8">Windows size (Advertisement window 16 bit)</td></tr>
<tr><td colspan="8">Checksum (16 bits)</td><td colspan="8">Urgent pointer (16 bits)</td></tr>
<tr><td colspan="16">Option (0-40 bits)</td></tr>
<tr><td colspan="16">Data optional</td></tr>
</table>

Source port: the source/transmitting application's port number.
Destination port: the port number of the receiving application.
Sequence number: random value which starts the connection and is increased according to the number of the transmitted bytes.
Acknowledge number: starts with zero by the receiving side and is increased according to the number of bytes received.
Header length/TCP data offset: the length of the header.
Control flags: help in managing data flow in specific situations.

- **URG/Urgent:** urgent pointer tells us that the data should be treated as priority over other data.
- **ACK/Acknowledgment:** to make sure that the receiver has received the data by sending an acknowledgment.
- **PSH/Push:** tells an application that the data should be transmitted immediately and we don't want to wait to fill the entire segment.

- **RST/Reset:** Reset the connection when an error occurs. This is not a normal way to finish the connection.
- **SYN/Synchronize:** start, synchronize the connection between sender and receiver by setting a sequence number for the connection.
- **FIN/Finish:** when the connection is terminated.

Window size: tells us how many bytes the receiver wants to receive.
Checksum: tells us if the header is damaged during connection.
Urgent pointer: used when the URG bit has been set, and it is used to indicate where the urgent data ends.
Option: These are optional fields for setting maximum segment sizes, selective acknowledgments and enabling window scaling for more efficient use of high-bandwidth networks.
Data: the application data that is transmitted.

Normal:

- Normal 3-way handshake (SYN-SYN/ACK-ACK)

  SYN → Seq: random number ex:1, ACK= 0
  SYN/ACK → Seq: random number ex:4, ACK= Seq+1, 1+1=2
  ACK → Seq: ACK=2, ACK= Seq+1, 4+1=5
- TCP 3-way handshake is between 2 hosts only.

Suspicious:

- Huge number of SYN messages in small time. (SYN scan)
- Usage of unusual flags.
- One host send to multiple ports or nodes. (scanning), (SYN flood).
- Usage of unusual port numbers ex:0 or 4444 (Metasploit)
- Using spoofed IP of one of 2 hosts to RST/Reset connection between them. Notice that the 2 hosts have different MAC addresses and one IP address. (TCP RST attack)
- Notice (TCP retransmission) which may points to (TCP Hijacking)

5- UDP

Structure:

Source port (16bit)	Destination port (16bit)
Length (16bit)	Checksum (16bit)
Data (32bit)	

Source port number: the port of the sender and it is considered to be the port to which replies can be sent if it is necessary. If it is not used, it is zero.
Destination port number: the destination port.
Datagram length: the whole size of the header and data.
Checksum: checks for errors and to provide integrity.
Data: the application data.

Normal:

It is used often over VOIP applications (skype, zoom) and gaming. Also with IOT services and any service which are connectionless does not care about orientation, So it does not have this so much vulnerabilities like TCP.

Suspicious:

- Huge number of UDP packets to multiple ports. (port scan)
- It could also be a sign for UDP Flooding which is a type of DOS attack. It is performed by sending huge number of UDP packets to different non used ports (which have no service listening on) of the server which forces it to respond with ICMP (destination unreachable) and this make the system is unavailable for legitimate clients.
  Notice that the attacker uses spoofed IP address as not to harm themselves by the huge number of the ICMP responds.
- Notice the URL requests with no reply. (C&C exfiltrate data)

6- DNS

Structure:

Offsets	Octet	0	1	2	3
Octet	Bit	0–7	8–15	16–23	24–31
0	0	DNS ID Number		QR | OpCode | AA | TC | RD | RA | Z	RCode
4	32	Question Count		Answer Count	
8	64	Name Server (Authority) Record Count		Additional Records Count	
12+	96+	Questions Section		Answers Section	
		Authority Section		Additional Information Section	

DNS ID Number: unique number for DNS query and its response
Query/Response (QR): tells us if this packet is a DNS query or response
OPCODE: defines the type of query
Authoritative Answers (AA): If this is set in a response packet, it indicates the response is from a name server with authority over the domain.
Truncation (TC): indicates the response was truncated because it is large.
Recursion Desired (RD): If this is set in a query, it indicates the DNS client requests a recursive query if the target name server doesn't contain the requested information.
Recursive Available (RA): If this is set in a response, it indicates the name server supports recursive queries.
Reserved (Z): defined to be set as all 0s, or as an extension of RCode.
Response Code (RCode): tells us if there are any errors in DNS response.
Question Count / Number of questions: the number of entries in the Question Section.
Answer Count / Number of answer RRs: The number of entries in the Answer Section.
Name Server (Authority) Record Count / Number of Authority RRs: The number of name server resource records in the Authority Section.

Additional Records Count / Number of Additional RRs: The number of other resource records in the Additional Information Section.
Questions Section: contains one or more queries, each query includes:
1) name field: contains the name that is being queried.
2) type field: indicates the type of question being asked about the name
Answers Section: In a reply packet, it contains one or more RRs that answer queries. A reply can return multiple RRs
Authority Section: contains RRs that point to authoritative name servers that can be used to continue the resolution process
Additional Information Section: contains RRs that hold additional information related to the query that is not absolutely necessary to answer the query.

Normal:

- Using UDP port **53.**
- DNS request and reply have the same Transaction \ DNS ID.
- DNS query is sent to DNS server.
- DNS replies with its requested record.

NOTE:
Recursion: the process of a DNS server queries another DNS server on behalf of the client to find an IP address, acting like a client. This is done when the first server did not find the answer for the request. And this recursive request has a different transaction ID.

Suspicious:

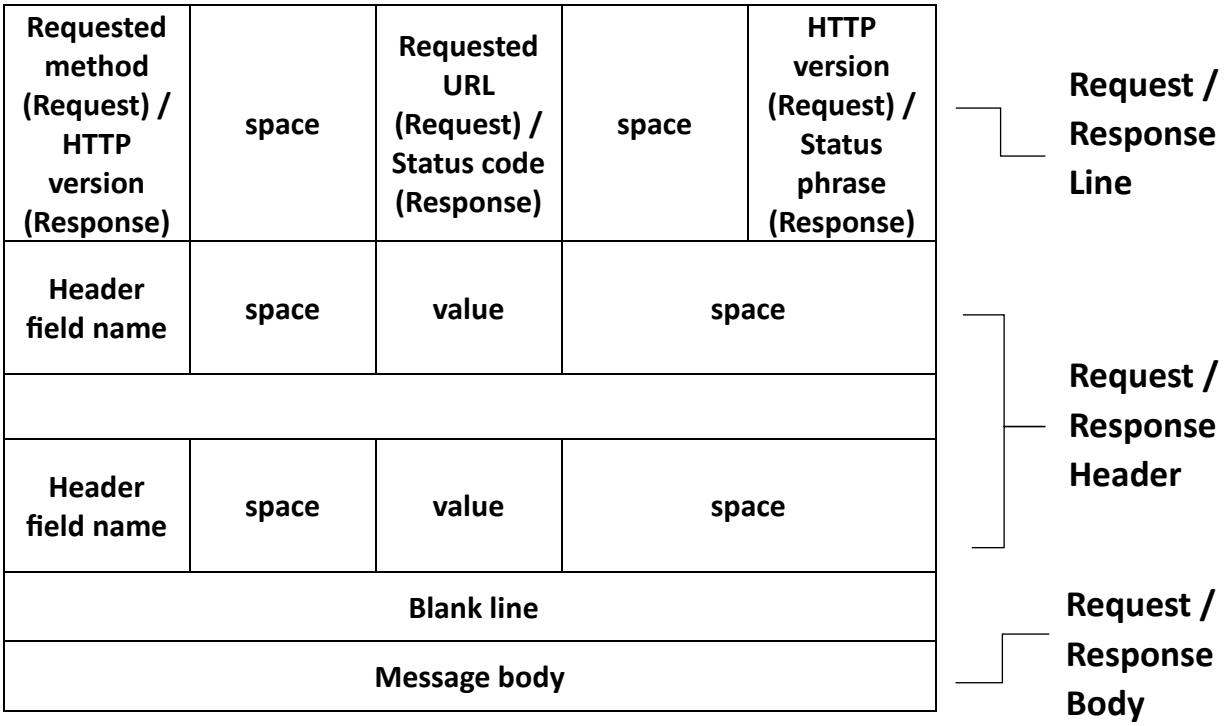
- Many DNS requests with no reply or vice versa
- Using TCP on port 53 instead of UDP
  (DNS zone transfer from a suspicious IP address)

NOTE: Organizations usually have more than one DNS server which use them as redundancy and to keep a save copy of DNS records, So they need a total management for these servers as when they change any configuration in one server, they have to implement this change to the other, this what is called DNS zone transfer. When a secondary DNS server requests an update of records from the DNS primary server, it uses AXFR or IXFR to get these updates which uses TCP on port 53 as the respond usually too big to be implemented using UDP and this is the case when we see unique transaction\DNS ID for more than one respond. But this is vulnerable as no authentication or encryption used here. Because of that, organizations usually use trusted IP address only and must be monitored besides using signatures to stop IP spoofing or they block this feature.

- Notice the IP address of DNS server and domain name (fakeDNS)
- Notice the number and size of the query from the client when it is bigger than usual, this is pointer to exfiltration of data using DNS tunneling.
- unusual domain names could be C&C domain generated by malware.

7- HTTP

Structure:



Requested method: the type of requested needed to be sent to the web server. (GET, POST, DELETE, PUT, HEAD)

HTTP version: the version of http used (ex: HTTP 1.0, HTTP 1.1).

Requested URL: the URL requested by the client.

Status code: 3-digits tells us status of the response (ex: 200, 300, 400).

Status phrase: short description of the status code.

Request Header: tells us information about how data that is wanted from the web server such as the data in plaintext format. Ex:

- **Header field:** Accept **Value:** image/jpeg
- **Header field:** Accept-language **Value:** en-us
- **Header field:** user-agent **Value:** Mozilla/4.0
- **Header field:** Accept-Encoding **Value:** {coding type}
- **Header field:** Accept-Ranges **Value:** bytes

- **Header field:** Connection **Value:** {Keep-alive/close}

Response Header: additional information related to the response data and the server. Ex:

- **Header field:** Date **Value:** {date and time }
- **Header field:** Server **Value:** {server OS}
- **Header field:** Last-Modified **Value:** {date and time }
- **Header field:** Accept-Ranges **Value:** bytes
- **Header field:** Content-Length **Value:** {length}
- **Header field:** content-type **Value:** text/html

Request body: is an optional part if there is a requirement data from the client such as any input from the them. (search, comment, feedback, etc.)

Response body: includes the requested data by client from the web server.

Normal:

- HTTP uses TCP on port **80** or **8080** or **8088** at the web server.
- HTTP has no encryption and shows the data with plaintext format.
- HTTP shows the fully domain name with specific host. (FQDN: host name + domain name)

Suspicious:

- Using IP address instead of FQDN unless it is internal server.
- Notice if it's encrypted. This is pointer to something try to hide malicious code or web shell.
- Notice web attacks behavior such as SQLI, XSS, Code I, LFI-RFI, IDOR.
- Notice automated tools name such as SQLmap in user-agent strings.
- Notice the number of http request packages into the web server (DOS) or to any input from the client. (Brute force)
- Notice the response payload for malicious files and malwares.
- Notice no of the outbound traffic as it could point to data exfiltration.

- Be aware when notice unusual URL pattern or extensions such as .exe or .zip as could be downloading malware.
- Also if you at organization you must notice the unusual timing for using or browsing through HTTP.

While we are talking about HTTP, we must also mention HTTPS.

It is the secure version of HTTP using the power of **SSL/TLS** encryption methods to encrypt the data transferred between client and server and ensures privacy and authentication.
To understand how it works, we should notice the concepts of SSL/TLS.

8- SSL

structure:

Handshake protocol	Change cipher spec protocol (8 bit)	Alert protocol (16 bit)	HTTP
SSL record protocol			
TCP			
IP			

Handshake protocol: establish the connection between client (client web browser) and web server.

- The client sends a Client Hello message to the server to define its SSL version it can support, session ID and cipher suite which contains cryptography methods the client supports and the same parameters for Server Hello message which is sent from the server to the client.
- The server sends its certificate and encryption key to the client and requests for client authentication.
- The client checks the certificate validation and then it sends its digital certificate to the web server.
- The client uses server **public key** to encrypt its own **private key**, then the client sends the **encrypted private key** to the server which will use this private key to encrypt the data.
- The client sends the cipher functions to the server and ends the handshake from its side.
- Then the web server also sends his cipher algorithms and ends the handshake from its side.
- The data now is sent encrypted using the client **private key**.

Change cipher spec protocol: It is 1 byte has a value of 1, it copies pending state (if the handshake protocol has not completed) to current state to update its cipher suite.

Alert protocol: for reporting error detected from client or server. It consists of 8-bit Level and 8-bit Alert.

- 8-bit level: describes alert level (warning and Fatal).

(1:warning)	close notify
Bad certificate	un supported certificate
No certificate	certificate revoked
Certificate expired	
Certificate unknown	
(2:Fatal)	
Handshake failure	Bad record MAC
Decompression failure	unexpected message
Illegal parameters	

- 8-bit Alert: is the alert code number of the above messages

SSL record protocol: responsible for encrypted data transmission and encapsulation of the data sent by the higher layer protocols also to provide basic security services to higher layer protocols such as Confidentiality, integrity and authentication.

9- TLS

Steps:

Client Hello: sent from client to server to establish the connection and to provide its parameters.

Server Hello: sent from server to the client to provide its parameters.

- Record header: contain the content type (Handshake), version of TLS and length.
- Handshake protocol: contains Handshake message type (client / server Hello) and length.
- Client version: TLS version
- Client Random: 32-bit random data
- Session ID: ID of the session between client and server
- Cipher suites: contains a list of which cryptographic methods it will support for key exchange, encryption with that exchanged key.
- Compression methods: contains a list of which compression methods it will support
- Extensions lists: list of optional extensions which the client or server can use to take action or enable new features.

Server Certificate: the server provides its certificate containing the server hostname, public key used by server and a proof from a 3rd party that the owner of this hostname has the private key of this public key.

- Record header: contain the content type (Handshake), version of TLS and length.
- Handshake protocol: contains Handshake message type (certificate) and certificates length.
- Certificates**:** certificate length and the certificate sent by the server.

Server key exchange generation: The server calculates a private/public keypair for key exchange.

Server key exchange: the server sends its public key to the client**.**

- Record header: contain the content type (Handshake) and version of TLS and length.
- Handshake protocol: contains Handshake message type (server key exchange) and length.
- Type of encryption: public key and length

Server Hello done: The server indicates it's finished with its half of the handshake.

- Record header: contain the content type (Handshake) and version of TLS and length.
- Handshake protocol: contains Handshake message type (Server Hello done) and length.

Client key exchange generation: The client calculates a private/public keypair for key exchange.

Client key exchange: the client encrypts its private key (premaster secret) using server public key which had been authenticated by the certificate through the client web browser and send it to the server.

- Record header: contain the content type (Handshake) and version of TLS and length.
- Handshake protocol: contains Handshake message type (client key exchange) and length.
- Type of encryption: public key and length

Each side now can calculate encryption keys used by each side based on random number and public key of each side.

Server application data: The server can now decrypt the client private key and use it to encrypt the data.

Client application data: The client sends the data it wants to the server.

Client close notify: client sends an alert that it is closing the connection.

NOTE: SSL/TLS use asymmetric and symmetric encryption.

NOTE: TLS has more than one handshake type based on tis version.

10- HTTPS

After we understanding the basic ideas of SSH/TLS we can now identify that HTTPS is same as HTTP with a security special feature.

Normal:

- uses TCP on port 443 or 8443
- Data is encrypted.
- Domain names presents its FQDN as in HTTP.

Suspicious:

- Data is not encrypted.
- No SSL/TLS handshake which is the base of security transmission.
- SSL/TLS header should not be empty.
- Notice if the is any expired certificate or self-signed certificate or certificated issued by untrusted authorities.
- Notice the size of the packet transferred ass it could have a malware payload or suspicious files.
- Be aware of the hostname and make sure that the requested hostname is the same as the one in the certificate.
- User agent behavior such as using automated tools or generated from unexpected geographic locations.
- (Spurious Retransmission) messages which means that the client believe that the server has not received the message, so the client decides to send it back. this pointing to some kind of Ransomware attacks, also you may notice (Zerowindow)
- Notice any suspicious domain name connecting on port 443 as it could be a C&C server.