PassQueen
a.k.a dumpsqueen.com

Wireshark Certified Network Analyst Practice Exam

Wireshark WCNA

Version Demo

Total Demo Questions: 12

Total Premium Questions: 120

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Network Analysis	7
Topic 2, Network Communication Fundamentals	17
Topic 3, TCP/IP Communications	24
Topic 4, Network Security	8
Topic 5, Wireshark	64
Total	120

QUESTION NO: 1

Which Wireshark feature is used to make the process of following TCP Sequence/Acknowledgment numbers easier to interpret?

A. sequence number flagging

B. sequence number prediction

C. relative sequence numbering

D. actual sequence number interpretations

ANSWER: C

Explanation:

relative sequence numbering is correct. TCP sequence and acknowledgment values are 32-bit numbers that commonly begin at large, apparently arbitrary initial sequence numbers. Reading the raw values directly makes it harder to see the progression of a TCP conversation, such as the number of bytes sent, acknowledged, retransmitted, or missing. Wireshark's relative sequence numbering feature presents the first observed sequence number in each direction as zero and displays subsequent values relative to that starting point. For example, a segment containing the first payload bytes can be shown as sequence number zero, and an acknowledgment can be shown as the corresponding count of received bytes. This makes the packet list and TCP analysis fields substantially easier to follow while preserving the underlying protocol behavior. The setting applies to displayed interpretation; it does not alter the captured packet data or the actual sequence numbers carried on the wire. Wireshark documents this preference as "Relative sequence numbers," and notes that relative TCP sequence numbers are enabled by default. See the Wireshark User's Guide: Preferences and the Wireshark User's Guide: TCP Analysis.

QUESTION NO: 2

What is the purpose of creating Wireshark profiles?

A. create a customized method of name resolution

B. discover and test RSA keys for traffic decryption

C. customize Wireshark for more efficient analysis in specific environments

D. create a manageable database of packets for use in third-party programs

ANSWER: C

Explanation:

Creating Wireshark profiles lets an analyst save and switch between sets of Wireshark preferences tailored to particular analysis tasks, networks, protocols, or operational environments. A profile can preserve settings such as display filters, coloring rules, protocol-specific preferences, column layouts, enabled or disabled dissectors, and other configuration choices. This avoids repeatedly reconfiguring the interface and analysis behavior when moving, for example, between VoIP troubleshooting, wireless analysis, or application-protocol investigation. Profiles therefore make packet analysis more consistent and efficient: the analyst can select an established profile whose settings highlight the traffic and fields that matter for the current work. Wireshark's User's Guide describes profiles as a mechanism for managing different preference configurations and notes that the active profile determines where personal configuration files are stored and read. A profile may also be shared or copied when a team needs a common analysis setup, helping standardize workflows across analysts. See the Wireshark User's Guide: Configuration Profiles and the Preferences and configuration documentation.

QUESTION NO: 3

Wireshark can be used to capture, reassemble and playback encrypted VoIP conversations.

A. True

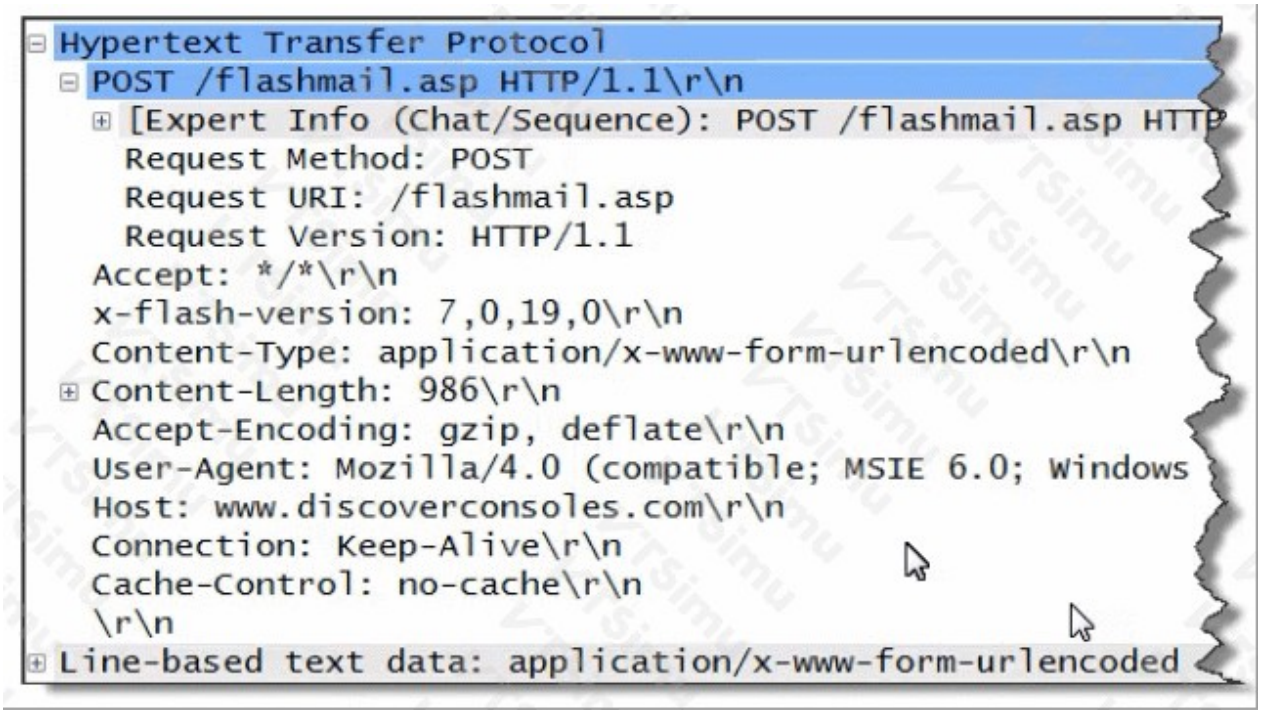
B. False

ANSWER: B

Explanation:

False is correct. Wireshark can capture packets carrying encrypted VoIP traffic, but merely capturing those packets does not make the voice payload available for reassembly, decoding, or playback. Encrypted RTP media, such as Secure RTP, protects the audio payload so that it cannot be interpreted by the RTP Player while it remains encrypted. To reconstruct and play a protected conversation, Wireshark must first have the applicable cryptographic material and support the negotiated encryption method. For example, SRTP analysis requires the relevant keying information to be available and configured so Wireshark can decrypt the media stream before normal RTP processing and codec decoding can occur. Without successful decryption, a capture may still reveal signaling, packet timing, endpoints, and other metadata, but it does not provide playable audio from the encrypted payload. This distinction is important in forensic and troubleshooting work: packet capture is possible regardless of encryption, whereas intelligible VoIP playback depends on access to the session's decryption keys and compatible dissector support. Wireshark's VoIP documentation describes RTP stream analysis and playback capabilities, while its SRTP documentation addresses the decryption prerequisites for protected media. See the Wireshark User's Guide: VoIP Calls and the Wireshark SRTP documentation.

QUESTION NO: 4



This packet was sent from an HTTP client.

A. True

B. False

ANSWER: A

Explanation:

True is correct. HTTP is an application-layer protocol in which a client initiates communication by sending a request to a server. In a packet trace, the direction is established from the TCP/IP addressing and port assignments: the client normally uses a dynamically selected ephemeral source port and sends traffic to the server's HTTP service port, commonly TCP port 80. The packet shown has the characteristics of traffic originating at that client endpoint and directed toward the HTTP service, so it is correctly identified as having been sent by an HTTP client.

Wireshark can dissect HTTP traffic and display the request details in its packet tree when the payload is available and the protocol is recognized. TCP stream context is also useful: an HTTP request appears in the client-to-server direction of the conversation, while the corresponding response travels from the server back to the client. This directional interpretation is fundamental when using Wireshark to distinguish request-generating hosts from web servers in a capture. See the Wireshark display-filter reference for HTTP filtering and the Wireshark User's Guide discussion of following streams for examining the bidirectional TCP conversation.

QUESTION NO: 5

Wireshark's Export feature can be used to identify HTTP objects and reassemble them into their original format.

A. True

B. False

ANSWER: A

Explanation:

True is correct. Wireshark's *File → Export Objects → HTTP* feature presents objects transferred in HTTP traffic, including such items as HTML pages, images, scripts, documents, and other response content. The Export Objects window identifies the available objects and provides metadata such as the host, request URI, content type, and content length, helping an analyst locate the item of interest in a capture. An object can then be selected and saved, allowing its payload to be recovered as a file for examination in the format indicated by the HTTP transfer metadata or file content. This is particularly useful when investigating downloaded files, reconstructing web content, or extracting evidence from unencrypted HTTP sessions. Wireshark also supports TCP stream reassembly, which enables higher-layer dissectors such as HTTP to process application data that was divided across multiple TCP segments. Consequently, an HTTP object does not need to be contained in one packet for the export facility to recover it. The Wireshark User's Guide documents the Export Objects dialogs and their purpose at Exporting Objects. Details on how Wireshark reassembles fragmented higher-layer protocol data are available in the Packet Reassembly section of the guide.

QUESTION NO: 6

The gratuitous ARP process is not required if a host is configured with a static IP address.

A. True

B. False

ANSWER: B

Explanation:

False is correct. A gratuitous ARP (GARP) is an unsolicited ARP announcement in which a host advertises its own IPv4-to-MAC-address mapping. Whether an address was assigned manually or obtained dynamically does not eliminate the value of this process. A host using a static address may send ARP probes or announcements when an interface becomes active, when its address changes, or when it needs to detect and defend against an address conflict. These messages can reveal that another device is already using the selected IPv4 address, helping prevent duplicate-address communication failures. They also allow peer devices to refresh stale ARP-cache entries after a host's network adapter, MAC address, or attachment point changes. In packet analysis, such traffic commonly appears as an ARP request or reply involving the sender's own protocol address; it is not limited to DHCP environments. The IPv4 Address Conflict Detection specification defines probing and announcement behavior for detecting conflicts and updating other hosts' cached address mappings, while Wireshark's ARP protocol reference identifies ARP as the mechanism that maps IPv4 addresses to link-layer addresses. See RFC 5227: IPv4 Address Conflict Detection and Wireshark ARP protocol reference.

QUESTION NO: 7

DNS responses contain four sections: Question, Answer RR, Authority RR and Additional RR.

A. True

B. False

ANSWER: A

Explanation:

True is correct. The DNS wire-message format defines four logical sections following the fixed header: Question, Answer, Authority, and Additional. The header contains separate count fields for each of these sections, allowing a response to state how many resource records or questions are present in each applicable section. The Question section normally repeats the queried name, type, and class so that a resolver can associate the reply with its query. The Answer section carries resource records that directly answer the question. The Authority section commonly identifies authoritative name servers or supplies other authority-related records, while the Additional section can include useful related records, such as address records for referenced name servers. A particular response does not need to contain records in every section; a section may have a count of zero. Nevertheless, these four sections are part of the defined DNS message structure, which is why the statement is accurate. The original DNS specification describes this layout in its message-format section, and the current DNS terminology document retains the same section names and roles. See RFC 1035, section 4.1 and RFC 9499.

QUESTION NO: 8

How do you quickly spot large gaps in time between packets in a trace file containing 6,000 packets?

A. Open and examine Wireshark's Expert Infos window.

B. Look for packets colorized based on Wireshark's default coloring rule for high delta times.

C. Set the Time column to Seconds Since Beginning of Capture and scroll through the trace file.

D. Set the Time column to Seconds Since Previously Displayed Packet and sort the Time column.

ANSWER: D

Explanation:

Set the Time column to Seconds Since Previously Displayed Packet and sort the Time column. This time-display format calculates each packet’s timestamp relative to the packet immediately preceding it in the currently displayed packet list. Consequently, a lengthy idle interval appears as a large value in the Time column on the packet that follows the gap. Sorting that column places the largest inter-packet delays together, allowing an analyst to identify significant pauses rapidly rather than manually inspecting thousands of timestamps in capture order. The word “displayed” is important: if a display filter is active, the delta is calculated against the previous packet that remains visible after filtering. This makes the method useful both for a complete capture and for focused analysis of a specific protocol, endpoint, or conversation. Wireshark documents the available timestamp display formats, including the delta relative to the previous displayed packet, in its User’s Guide. Its packet-list documentation also describes sorting by clicking a column heading, which is the practical step that turns the delta values into a quick ranked view of the largest gaps. See the Wireshark User’s Guide: Time Display Formats and Wireshark User’s Guide: Packet List.

QUESTION NO: 9

By default, Wireshark will only dissect port 443 traffic as SSL/TLS traffic. If you are using another port for SSL/TLS communications, you must add that port number in the HTTP preferences setting for SSL/TLS ports.

A. True

B. False

ANSWER: B

Explanation:

False is correct because SSL/TLS protocol identification and port assignment are controlled by Wireshark's TLS dissector, not by the HTTP dissector's preferences. Wireshark can recognize TLS through its normal TCP port-based dissector bindings, and users can configure additional TCP ports for TLS in the TLS protocol preferences when a service uses TLS on a nonstandard port. The relevant preference is commonly displayed as a TCP ports setting under *Protocols* → *TLS*; its exact default list can vary by Wireshark version and installed dissector registrations. This makes the statement inaccurate in two important respects: it is not generally limited exclusively to port 443, and HTTP preferences are not the location for configuring TLS ports. When traffic uses a port that is not automatically associated with TLS, an analyst can add the port in the TLS preferences or use *Decode As* to force Wireshark to decode that conversation as TLS. Wireshark's TLS documentation describes TLS dissection and configuration, while the User's Guide documents the Decode As mechanism for assigning a dissector to traffic that uses an unusual port. See Wireshark Wiki: TLS and Wireshark User's Guide: Protocol Dissection and Decode As.

QUESTION NO: 10

Which traffic characteristic is often seen when analyzing database applications that transfer individual records across the network?

A. small packet sizes

B. multicast responses

C. large delays between transmissions

D. separate connections for each record

ANSWER: A

Explanation:

small packet sizes is correct because database clients and servers commonly exchange discrete requests, status messages, and individual result records rather than always sending a continuous bulk data stream. When each record or operation is transmitted separately, the application payload in each exchange may be relatively small. This pattern can be especially apparent in request/response database workloads, where a client submits a query or fetch request and the server returns records incrementally. At the transport level, these small application writes are often visible as many short TCP segments, although TCP buffering, offloading, encryption, and protocol framing can affect exactly how the packets appear in a capture. Wireshark supports evaluating this behavior by exposing each frame's captured length and protocol fields in the Packet Details pane, and by summarizing packet-size distributions with packet-length statistics. Looking at those distributions alongside a database conversation helps an analyst recognize workloads dominated by many small record-oriented exchanges rather than large bulk transfers. See the Wireshark User's Guide: Packet Details and the Packet Lengths statistics documentation.

QUESTION NO: 11

When you apply a display filter, the Status Bar indicates the total number of packets captured and the packets displayed.

A. True

B. False

ANSWER: A

Explanation:

True is correct. A Wireshark display filter changes which packets are shown in the packet list; it does not alter or remove packets from the capture file or from Wireshark's underlying captured-packet set. The Status Bar provides packet-count information that lets an analyst compare the total packet count with the number currently displayed after the filter is evaluated. This immediate feedback is useful for confirming that a filter has taken effect and for judging how much of the original traffic remains in view. For example, a capture may contain thousands of packets in total, while a filter such as `dns` limits the packet list to only packets matching the DNS protocol. Wireshark documents display filters as a mechanism for controlling the packets displayed in the packet list, separately from capture filters, which determine what is captured in the

first place. The packet-count indicators therefore support the distinction between the full capture and the filtered display. See the Wireshark User's Guide: Building Display Filter Expressions and the Wireshark User's Guide: Main Window.

QUESTION NO: 12

Applications may override the default port value defined in the TCP/IP stack services file.

A. True

B. False

ANSWER: A

Explanation:

True is correct. A TCP or UDP port is selected by the application when it creates and binds a socket. The system `services` file is principally a local name-to-port mapping: it associates familiar service names, such as `http`, with conventional transport ports. It does not reserve those ports exclusively for a protocol or prevent an application from listening on another permitted port. Consequently, software can be configured to listen on a nondefault port, and clients can be configured to connect to that same port. This is common for administration interfaces, development environments, multiple instances of a service on one host, and deployments where the conventional port is unavailable or intentionally avoided. In packet analysis, Wireshark may initially use port-based conventions to help identify traffic, but protocol dissection can also depend on packet contents, decoding preferences, and explicit user configuration. Wireshark's TCP protocol documentation describes TCP as carrying application data between port endpoints: Wireshark TCP display-filter reference. The assigned, conventional service-port registry is maintained by IANA: IANA Service Name and Transport Protocol Port Number Registry.