سوال 1:

Wiresharkچگونه بسته هاي TCPو UDPرا شناسايي و تفكيک ميكند؟ چه فيلترهايي براي تحليل اين بسته ها مفيد هستند؟

جواب :

Wireshark با استفاده از **فیلدهای پروتکل** در هدر بسته‌ها، پروتکل انتقال TCP یا UDP را شناسایی می‌کند. هر بسته در لایه Transport لایه ۴ مدل OSI شامل اطلاعاتی است که نوع پروتکل را مشخص می‌کند:

شناسایی TCP وUDP :

- **TCP:** در هدر آن فیلدهایی مثل:
  Sequence Number، Acknowledgment Number و Flags (SYN, ACK, FIN) و ... وجود دارد.
- **UDP:** هدر بسیار ساده‌تر است و فقط شامل Source Port، Length Destination Portو Checksum است.

فیلترهای مفید درWireshark :

- **فیلترهای پایه**:
  - tcp: فقط بسته‌های TCP را نمایش می‌دهد.
  - udp: فقط بسته‌های UDP را نمایش می‌دهد.
- **فیلتر بر اساس پورت**:
  - tcp.port == 80: ترافیک TCP روی پورت 80 مثل HTTP
  - udp.port == 53 ترافیک UDP روی پورت 53 مثل DNS
- **فیلتر بر اساس وضعیت ارتباط**:**TCP**
  - tcp.flags.syn == 1: فقط بسته‌های **SYN** شروع ارتباط.
  - tcp.flags.reset == 1: بسته‌های **RST**اتصال قطع شده.
  - tcp.analysis.retransmissio: بسته‌های **Retransmitted**یا ارسال مجدد.
- **فیلتر ترکیبی**:
  - tcp and ip.src == 192.168.1.1: فقط ترافیک TCP از یک IP خاص.
  - udp and frame contains "google": ترافیک UDP که محتوای کلمه "google" است.

سوال 2:

چگونه میتوان با استفاده از Wireshark تاخیرLatency و زمان پاسخ Response Time در یک ارتباط TCPرا اندازه گیري کرد؟

جواب:

اندازه‌گیری(Round-Trip Time (RTT :

RTT مدت زمانی است که طول می‌کشد تا یک بسته از مبدأ به مقصد برود و پاسخ آن برگردد.

- استفاده از فیلد tcp.analysis.ack_rtt: به‌صورت خودکارRTT را برای بسته‌هایTCP محاسبه میکند.
- ستون RTT را در لیست بسته‌ها اضافه کنید .(راست‌کلیک → Column Preferences → Add → RTT)
- از Statistics → TCP Stream Graphs → Round Trip Time استفاده کنید.

اندازه‌گیری Response Time زمان پاسخ سرور:

زمان پاسخ، مدت زمانی است که سرور برای پردازش درخواست و ارسال پاسخ نیاز دارد. مثال: درHTTP ، زمان بین GET و HTTP 200 OK است.

- نمونه فیلتر برای اینکار (http.request || http.response) && http و محاسبه زمان پاسخ – زمان درخواست
- رسم نمودار پاسخ با استفاده از Statistics → IO Graph با فیلتر http.response

سوال 3:

تفاوت هاي اصلی بين پروتكل هاي TCPو UDPچيست و چگونه بر تحليل ترافيک آنها در Wireshark تاثير ميگذارد؟

جواب:

مقایسه کلی TCP وUDP :

ویژگی	*TCP* *(Transmission Control Protocol)*	*UDP* *(User Datagram Protocol)*
اتصالگرا	3-way handshak	بدونhandshake
قابلیت اطمینان	تضمین تحویل داده‌ها	بدون تضمین تحویل
ترتیب داده‌ها	مرتب شدن باSequence Number	بدون ترتیب
کنترل ازدحام	مثل الگوریتم‌های(Tahoe, Reno)	بدون کنترل ازدحام
سرعت	کندتر به دلیل سربار	سریع‌تر
هدر	بزرگ‌تر (حداقل ۲۰ بایت)	کوچک‌تر (۸ بایت)
مصارف رایج	HTTP, FTP, SSH و ایمیل	DNS, VoIP بازی‌های آنلاین وStreaming

تأثیر تفاوت‌ها بر تحلیل ترافیک درWireshark

الف) تحلیل TCP درWireshark:

- دارای شماره توالی (Sequence Number) و تأییدیه (ACK) است؛ بنابراین Wireshark می‌تواند جریان‌های TCP را به خوبی دنبال و بازسازی کند.
- قابلیت تحلیل دقیق انتقال داده، تشخیص تأخیر، بسته‌های ازدست‌رفته یا مجدداً ارسال‌شده را دارد.
- سه مرحله آغاز اتصال (Three-Way Handshake) و پایان آن (Four-Way Termination) مشاهده میشود.
- Wiresharkمی‌تواند Stream Followرا برای بازسازی جریان کامل یک ارتباط TCP انجام دهد.

ب) تحلیل UDP درWireshark:

- به دلیل نبود مکانیزم ارتباطی Wireshark فقط می‌تواند بسته‌های جداگانه را نمایش دهد.
- بازسازی جریان یا تشخیص ازدست‌رفتن بسته‌ها ممکن نیست.
- هدر ساده‌تر و اطلاعات کمتری برای تحلیل وجود دارد.
- درکاربردهایی مانند DNS یا پخش ویدئو، فقط می‌توان بررسی کرد که آیا درخواست و پاسخ رد و بدل شده‌اند یا نه.

سوال 4:

چگونه میتوان با استفاده از ،Wiresharkمشکلاتی مانند دوباره ارسال بسته ها یا Retransmissions بسته هاي گم شده (Lost packets) مشکلات جریان یا Flow Control در ارتباطات TCP را شناسایی و تحلیل کرد؟

جواب:

شناسایی دوبارهارسال بستهها(TCP Retransmissions) :

در پروتکلTCP ، اگر یک بسته به مقصد نرسد یا تأییدیه (ACK) آن دریافت نشود، فرستنده آن بسته را دوباره ارسال می‌کند. Wireshark به‌صورت خودکار این بسته‌های تکراری را تشخیص می‌دهد و در ستون **"Info"** آن‌ها را با برچسب‌هایی مانند: TCP Retransmission ، TCP Fast Retransmission ، TCP Spurious Retransmission نمایش می‌دهد.

برای بررسی بهتر میتوانیم از فیلتر زیر استفاده کنیم:

tcp.analysis.retransmission: با این فیلتر می‌توانید فقط بسته‌های دوباره‌ارسال‌شده را ببینیم و محل مشکل را پیدا کنیم.

شناسایی بسته‌های گم‌شده(Lost Packets) :

وقتی گیرنده منتظر بسته‌ای است و آن را دریافت نمی‌کند، ولی بعد از مدتی دوباره‌ارسالی از سوی فرستنده انجام می‌شود، این نشان‌دهنده **بسته گم‌شده در مسیر** است. Wiresharkنمی‌تواند بسته‌ای که **اصلاً دریافت نشده** را نشان دهد، اما می‌تواند آن را **حدس بزند** بر اساس شماره‌های توالی (Sequence Numbers) یا برچسب:

- TCP Previous Segment Not Captured

همچنین می‌توانیم از فیلتر زیر برای شناسایی بسته‌های از دست‌رفته استفاده کنیم: tcp.analysis.lost_segment

شناسایی مشکلات جریان(Flow Control) :

درTCP ، کنترل جریان باعث می‌شود فرستنده بیش از حد توان گیرنده، داده ارسال نکند. دو مفهوم مهم در این زمینه:

- **Window Size اندازه پنجره :**نشان می‌دهد گیرنده چه مقدار داده می‌تواند بدون ACK دریافت کند.
- اگر این مقدار صفر شود، ارتباط کند یا متوقف می‌شود. Wireshark این وضعیت را با برچسب:
  - Zero Windowنشان می‌دهد.

فیلتر مفید: tcp.analysis.zero_window

ابزارهای مفید در Wireshark برای تحلیل این مشکلات:

- **Follow TCP Stream:** برای مشاهده مکالمه کامل بین دو طرف.
- **Statistics > TCP Stream Graphs > Time-Sequence Graph (tcptrace style)**: برای دیدن زمان‌بندی و ترتیب بسته‌ها.
- **Expert Info (Analyze > Expert Information)**: برای مشاهده هشدارهای تحلیل خودکار Wireshark درباره مشکلاتی مانندRetransmission ، Lost Segment و غیره.