

Proyecto:	Seguridad Addin Sip-R / Sip-S
Elaborado por:	Angelo Bendezu L.
Fecha	07/05/2025

INFORME DE PRUEBAS

Addin Sip-R y Sip-S

Conexión NET: TCP

1. Introducción

- **Objetivo del informe:** Explicar los detalles de la configuración de la conexión entre el **Add-in de Excel** y el **servicio WCF** usando el protocolo net.tcp sobre el puerto 84.
- **Resumen:** Descripción general del sistema involucrado, los roles del Add-in y del servicio WCF, y el protocolo utilizado (net.tcp).

2. Descripción del escenario

- **IP Origen:** 172.16.8.3 (Add-in de Excel).
- **IP Destino:** 172.16.8.3 (Servidor que hospeda el servicio WCF).
- **Puerto:** 84 (Puerto usado para la conexión net.tcp).
- **Protocolo de comunicación:** net.tcp.
- **Versión de WCF:** Mención de la versión de WCF utilizada.
- **Tipo de servicio WCF:** Descripción de cómo el servicio WCF está configurado (por ejemplo, uso de netTcpBinding, configuraciones de seguridad, etc.).

3. Configuración técnica del servicio WCF

- **Binding Configurado:** Detalle de la configuración de netTcpBinding en el WCF, incluyendo:
 - **Seguridad:** ¿Está habilitada la seguridad de transporte (Transport) o de mensaje?
 - **Timeouts:** Configuración de tiempos de espera (sendTimeout, receiveTimeout, etc.).
 - **Tamaño de buffer:** Configuración de maxBufferPoolSize, maxBufferSize, etc.
 - **Otros parámetros de configuración:** maxConnections, maxReceivedMessageSize, etc.

Proyecto:	Seguridad Addin Sip-R / Sip-S
Elaborado por:	Angelo Bendezu L.
Fecha	07/05/2025

4. Configuración de seguridad

- **Método de seguridad:**
 - Descripción de si se está usando seguridad a nivel de **transporte** (Transport) o a nivel de **mensaje** (Message).
 - **Autenticación:** ¿Se está utilizando autenticación de tipo **Windows** o **Certificados**?
 - **Cifrado:** ¿Se está utilizando cifrado de datos en tránsito? Si se usa Transport, ¿es posible que esté implementado a través de SSL/TLS?
 - **Cliente y Servidor:** Descripción de las credenciales configuradas para el cliente y el servidor.

5. Configuración en el cliente (Add-in de Excel)

- **Código de configuración del cliente:** Incluye los fragmentos de código utilizados para configurar la conexión netTcpBinding en el cliente.
 - **Configuración de seguridad:** Asegurarse de que la configuración del cliente coincide con la del servicio WCF, como las credenciales y el tipo de seguridad.
 - **Gestión de tiempo de espera:** Descripción de los valores de openTimeout, sendTimeout, y receiveTimeout configurados en el cliente.
 - **Gestión de errores:** Explicar cómo se gestionan los errores y excepciones en la comunicación.

6. Análisis de rendimiento

- **Impacto del puerto 84:** Descripción de si el uso de un puerto específico (84) tiene alguna implicación en el rendimiento o si requiere configuraciones adicionales en cortafuegos/routers.

7. Seguridad de la comunicación

- **Cifrado de datos en tránsito:** Si se está usando net.tcp con seguridad de **transporte**, los datos están cifrados en tránsito. Explicar cómo se implementa el cifrado.
- **Autenticación:** Verificar si el servicio está configurado para realizar autenticación en los clientes y cómo se asegura que el cliente es quien dice ser.
- **Integridad de los mensajes:** ¿Se asegura la integridad de los mensajes entre el cliente y el servidor? ¿Cómo se protege contra modificaciones no autorizadas?

Proyecto:	Seguridad Addin Sip-R / Sip-S
Elaborado por:	Angelo Bendezu L.
Fecha	07/05/2025

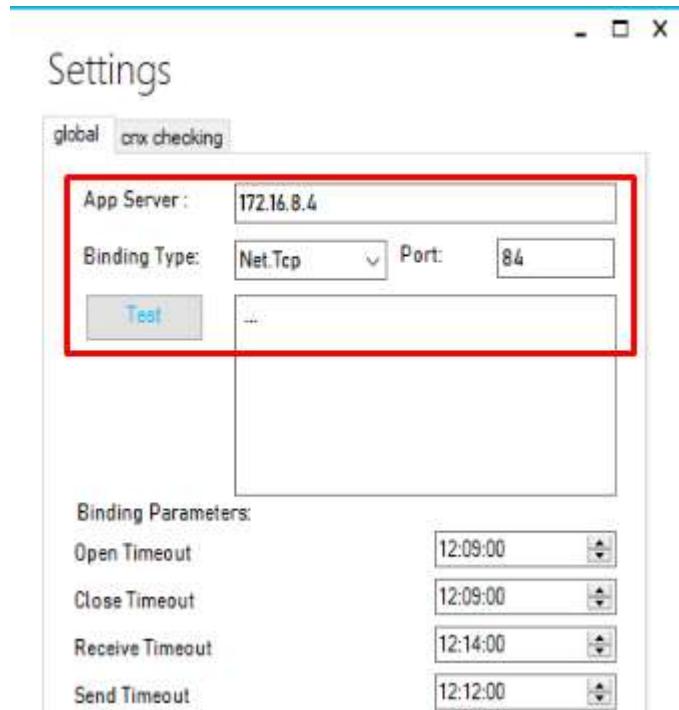
8. Análisis de Wireshark

- **Captura de tráfico:** Detalle de las capturas realizadas en **Wireshark** para verificar la seguridad de la conexión.
 - **Filtrado de tráfico(net:tcp):** Cómo se filtraron los paquetes de red para enfocarse solo en los de net.tcp sobre el puerto 84.

El procedimiento indicado considera lo siguiente:

- Instalación de la herramienta WireShark en el cliente (equipo donde está instalado el Addin)
- Se revisan los paquetes enviados y se verifica si están cifrados para corroborar la seguridad en la conexión.

ESCENARIO ANTERIOR (SIN SEGURIDAD)

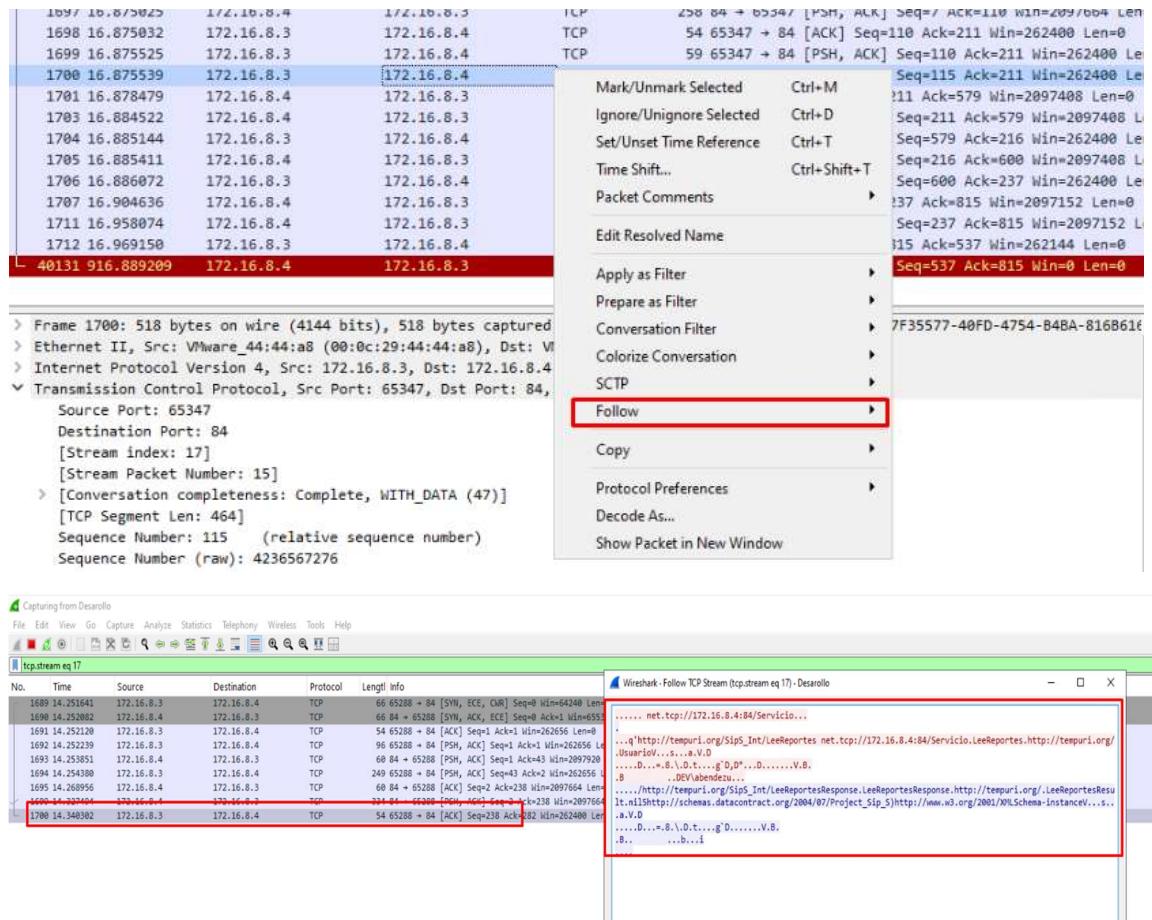


Se filtra en Wireshark los paquetes enviados hacia al Servidor Sip-S.

Para ello, usaremos el comando:

```
ip.addr == 172.16.8.3 && tcp.port == 84
```

Proyecto:	Seguridad Addin Sip-R / Sip-S
Elaborado por:	Angelo Bendezu L.
Fecha	07/05/2025



Como se puede apreciar en la imagen, al ver los paquetes, estos son legibles, por lo tanto, podemos indicar que los paquetes no tienen cifrado.

ESCENARIO NUEVO (CON SEGURIDAD)

Nos aseguramos que el servicio WCF tenga la configuración de seguridad, para ello revisamos el archivo config.

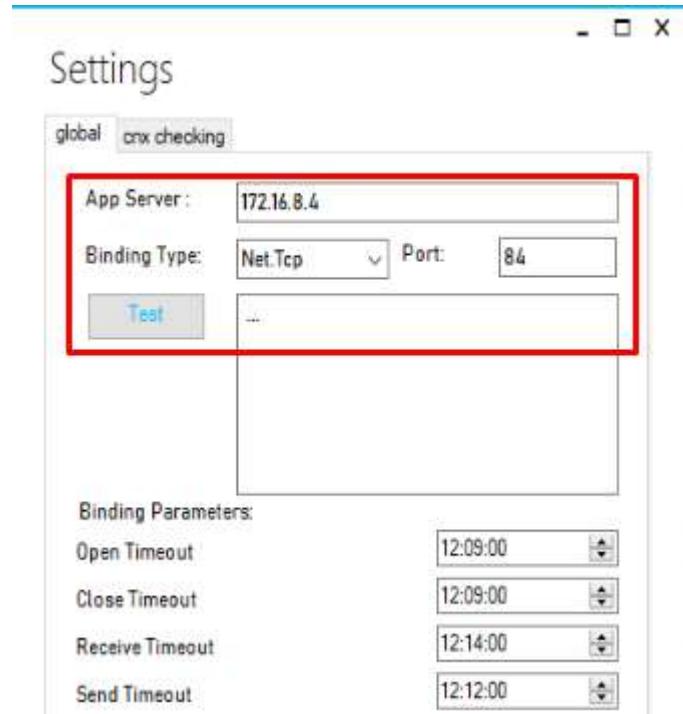
```

<system.serviceModel>
  <bindings>
    <netTcpBinding>
      <binding name="netTcpBinding" receiveTimeout="00:15:00" sendTimeout="00:15:00" closeTimeout="00:10:00" openTimeout="00:10:00"
        <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" maxBytesPerRe
        <reliableSession ordered="true" inactivityTimeout="00:10:00" enabled="false"/>
        <security mode="None">
          <transport clientCredentialType="Windows" protectionLevel="EncryptAndSign"/>
        </security>
      </binding>
    </netTcpBinding>
    <wsHttpBinding>
  
```

Este es el primer paso para asegurar que el WCF exige seguridad al cliente, de lo contrario la conexión no viajará cifrada.

Repetimos, ahora el mismo paso con el nuevo componente Sip-R:

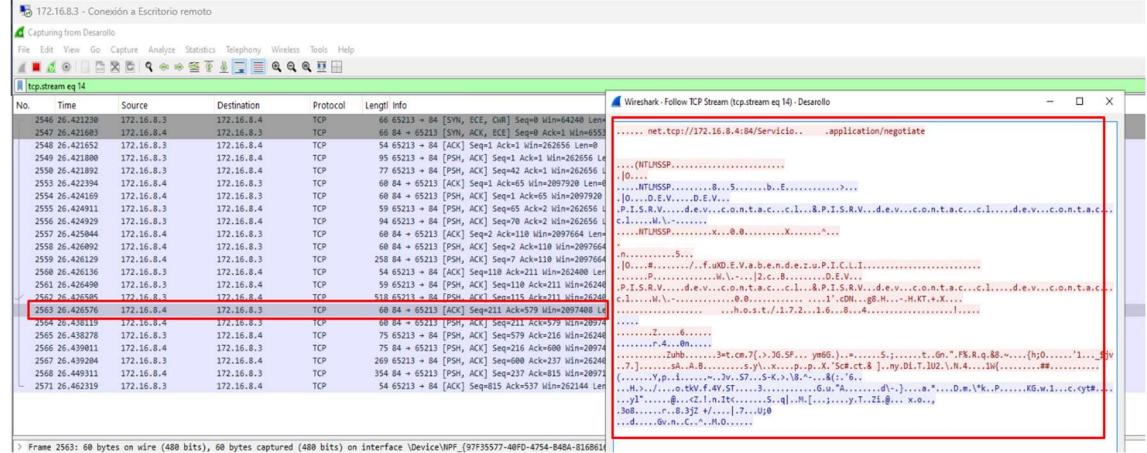
Proyecto:	Seguridad Addin Sip-R / Sip-S
Elaborado por:	Angelo Bendezu L.
Fecha	07/05/2025



Se filtra en Wireshark los paquetes enviados hacia al Servidor Sip-S.

Para ello, usaremos el comando:

```
ip.addr == 172.16.8.3 && tcp.port == 84
```



No.	Time	Source	Destination	Protocol	Length	Info
2546	26.421220	172.16.8.3	172.16.8.4	TCP	66	[SYN, ECN, QM] Seq=0 Win=64240 Len=40
2547	26.421683	172.16.8.4	172.16.8.3	TCP	66	84 + 65213 [SYN, ACK, ECN] Seq=1 Ack=1 Win=65535
2548	26.421652	172.16.8.3	172.16.8.4	TCP	54	65213 + 84 [ACK] Seq=1 Ack=1 Win=65535 Len=0
2549	26.421682	172.16.8.4	172.16.8.3	TCP	95	65213 + 84 [ACK] Seq=1 Ack=1 Win=65535 Len=0
2550	26.421692	172.16.8.3	172.16.8.4	TCP	77	65213 + 84 [PSH, ACK] Seq=2 Ack=1 Win=65535
2553	26.422394	172.16.8.3	172.16.8.4	TCP	68	84 + 65213 [ACK] Seq=3 Ack=65 Win=2097520 Len=0
2554	26.424169	172.16.8.4	172.16.8.3	TCP	68	84 + 65213 [PSH, ACK] Seq=1 Ack=65 Win=2097520 Len=0
2555	26.424011	172.16.8.3	172.16.8.4	TCP	59	65213 + 84 [PSH, ACK] Seq=65 Ack=2 Win=2097520 Len=0
2556	26.424029	172.16.8.3	172.16.8.4	TCP	94	65213 + 84 [PSH, ACK] Seq=78 Ack=2 Win=2097520 Len=0
2557	26.425944	172.16.8.3	172.16.8.4	TCP	68	84 + 65213 [ACK] Seq=Q Ack=116 Win=2097520 Len=0
2558	26.426092	172.16.8.4	172.16.8.3	TCP	68	84 + 65213 [PSH, ACK] Seq=2 Ack=110 Win=2097520 Len=0
2559	26.426129	172.16.8.4	172.16.8.3	TCP	258	84 + 65213 [PSH, ACK] Seq=7 Ack=110 Win=2097520 Len=0
2560	26.426136	172.16.8.3	172.16.8.4	TCP	54	65213 + 84 [ACK] Seq=110 Ack=216 Win=262408 Len=0
2561	26.426498	172.16.8.3	172.16.8.4	TCP	59	65213 + 84 [PSH, ACK] Seq=216 Ack=211 Win=262408 Len=0
2562	26.426501	172.16.8.4	172.16.8.3	TCP	503	84 + 65213 [ACK] Seq=216 Ack=211 Win=262408 Len=0
2563	26.426516	172.16.8.4	172.16.8.3	TCP	68	84 + 65213 [ACK] Seq=211 Ack=217 Win=262408 Len=0
2564	26.426519	172.16.8.4	172.16.8.3	TCP	68	84 + 65213 [PSH, ACK] Seq=211 Ack=217 Win=262408 Len=0
2565	26.426578	172.16.8.3	172.16.8.4	TCP	75	65213 + 84 [PSH, ACK] Seq=5 Win=262408 Len=0
2566	26.439011	172.16.8.4	172.16.8.3	TCP	75	84 + 65213 [PSH, ACK] Seq=216 Ack=608 Win=28974
2567	26.439304	172.16.8.3	172.16.8.4	TCP	269	65213 + 84 [PSH, ACK] Seq=608 Ack=237 Win=262408 Len=0
2568	26.449311	172.16.8.4	172.16.8.3	TCP	354	84 + 65213 [PSH, ACK] Seq=237 Ack=815 Win=28974 Len=0
2571	26.462319	172.16.8.3	172.16.8.4	TCP	54	65213 + 84 [ACK] Seq=815 Ack=537 Win=262144 Len=0

Frame 2563: 68 bytes on wire (408 bits), 68 bytes captured (408 bits) on interface \Device\WIFI_{97F35577-40FD-4754-B48A-816961

Se puede observar los datos se encuentra en forma binaria, ello indica que la conexión esta cifrada.

Cifrado observado: En las capturas de Wireshark se observa que los paquetes están cifrados o no. Esto se puede verificar viendo si los paquetes son legibles (texto claro) o si se muestra el tráfico cifrado.

Proyecto:	Seguridad Addin Sip-R / Sip-S
Elaborado por:	Angelo Bendezu L.
Fecha	07/05/2025

CONCLUSIONES:

Identificar el tráfico:

- Como net.tcp no es HTTP, no verás GET, POST, ni TLS Handshake.
- Normalmente verás tráfico TCP normal.
- Para verificar si viaja cifrado o no:
 - Haz clic derecho en un paquete → Follow > TCP Stream.

Observaciones del Stream

Observación	Significa	Seguridad
Texto legible como XML, SOAP, nombres de métodos WCF	No hay cifrado	 No seguro
Basura, símbolos raros, datos binarios que no son comprensibles	Hay cifrado o empaquetado	 Seguro

- **Filtrado de tráfico (HTTPS):** Cómo se filtraron los paquetes de red para enfocarse solo en los de HTTPS sobre el puerto 8443.

El procedimiento indicado considera lo siguiente:

- Instalación de la herramienta Wireshark en el cliente (equipo donde está instalado el Addin)
- Se revisan los paquetes enviados y se verifica si están cifrados para corroborar la seguridad en la conexión.

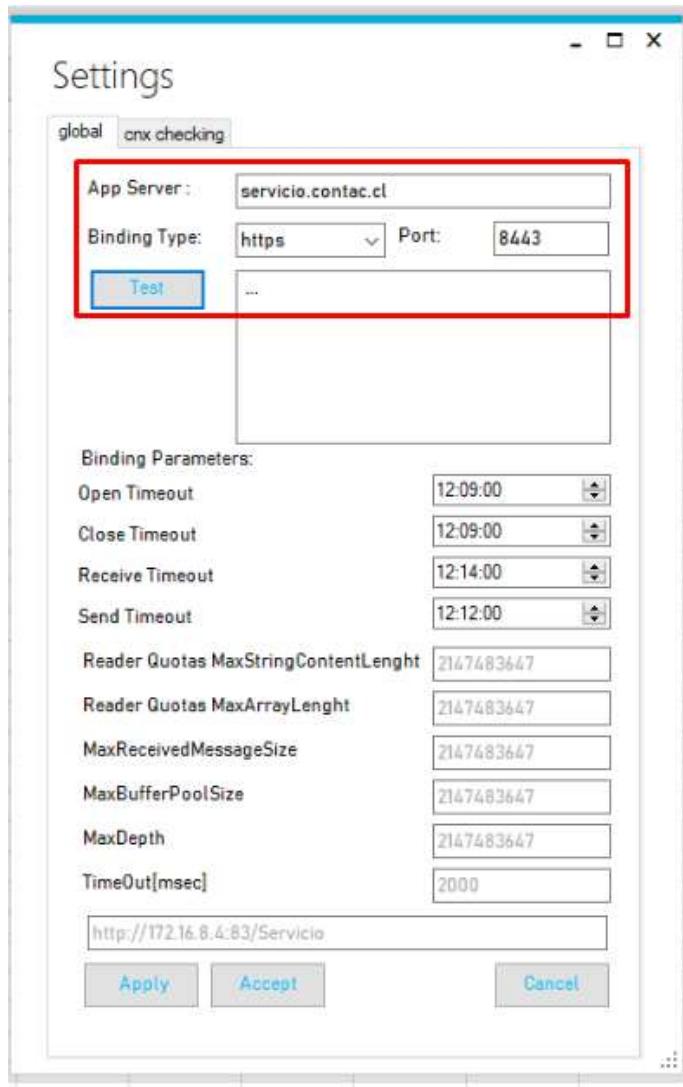
ESCENARIO ANTERIOR (SIN SEGURIDAD): No existe, existía conexión con HTTP Y HTTPS.

ESCENARIO NUEVO (CON SEGURIDAD)

Tener en cuenta las siguientes consideraciones:

- El puerto 8443 tiene que tener el certificado SSL.
- Recordar que para el llamado del servicio WCF, debe ser por el nombre del dominio, o en su defecto colocarlo en la red interna pero siempre con nombre del dominio.
- El servicio WCF siempre se despliega mediante Servicio Windows.

Activamos la conexión de Addin por la URL del servicio WCF:



Se filtra en Wireshark los paquetes enviados hacia al Servidor Sip-S.

Para ello, usaremos el comando:

```
ip.addr == 172.16.8.3 && tcp.port == 8443
```

tcp.stream eq 9						
No.	Time	Source	Destination	Protocol	Length	Info
1409	12.546049	172.16.8.3	172.16.8.4	TLSv1	304	Application Data, Application Data
1410	12.547459	172.16.8.4	172.16.8.3	TLSv1	144	Application Data, Application Data
1411	12.547545	172.16.8.3	172.16.8.4	TLSv1	688	Application Data, Application Data
1412	12.558531	172.16.8.4	172.16.8.3	TLSv1	816	Application Data, Application Data
1414	12.574578	172.16.8.3	172.16.8.4	TCP	54	62960 + 8443 [ACK] Seq=853 Ack=853 Win=8195 Len=0
7009	112.561095	172.16.8.3	172.16.8.4	TCP	54	62960 + 8443 [FIN, ACK] Seq=853 Ack=853 Win=8195 Len=0
7010	112.561906	172.16.8.4	172.16.8.3	TCP	60	8443 + 62960 [FIN, ACK] Seq=853 Ack=886 Win=8195 Len=0
7011	112.561936	172.16.8.3	172.16.8.4	TCP	54	62960 + 8443 [ACK] Seq=886 Ack=854 Win=8195 Len=0

Como se aprecia existe una conexión al protocolo indicado mediante TLS v1 , luego para ver los paquetes recibidos, vemos el detalle del Stream.

No.	Time	Source	Destination	Protocol	Length	Info
1409	12.546049	172.16.8.3	172.16.8.4	TLSv1	304	Application Data, Application Data
1410	12.547459	172.16.8.4	172.16.8.3	TLSv1	144	Application Data, Application Data
1411	12.547545	172.16.8.3	172.16.8.4	TLSv1	688	Application Data, Application Data
1412	12.558531	172.16.8.4	172.16.8.3	TLSv1	816	Application Data, Application Data
1414	12.574578	172.16.8.3	172.16.8.4	TCP	54	62960 → 8443 [ACK] Seq=885 Ack=853 Win=8195 Len=0
7009	112.561095	172.16.8.3	172.16.8.4	TCP	54	62960 → 8443 [FIN, ACK] Seq=885 Ack=853 Win=8195 Len=0
7010	112.561900	172.16.8.4	172.16.8.3	TCP	60	8443 → 62960 [FIN, ACK] Seq=853 Ack=886 Win=8195 Len=0
7011	112.561936	172.16.8.3	172.16.8.4	TCP	54	62960 → 8443 [ACK] Seq=886 Ack=854 Win=8195 Len=0

Mark/Unmark Selected Ctrl+M
 Ignore/Unignore Selected Ctrl+D
 Set/Unset Time Reference Ctrl+T
 Time Shift... Ctrl+Shift+T
 Packet Comments ▾
 Edit Resolved Name
 Apply as Filter ▾
 Prepare as Filter ▾
 Conversation Filter ▾
 Colorize Conversation ▾
 SCTP ▾
 Follow ▾ TCP Stream Ctrl+Alt+Shift+T
 Copy ▾ 10FD-4754-B4BA-816861609900
 Protocol Preferences ▾
 Decode As...
 Show Packet in New Window

Al ver los paquetes se observa texto binario, lo cual indica que la conexión está viajando cifrado.

Proyecto:	Seguridad Addin Sip-R / Sip-S
Elaborado por:	Angelo Bendezu L.
Fecha	07/05/2025

Recomendación práctica:

- **HTTP** en un puerto alternativo como 83 está bien si no usas el 80 (evita conflictos con otros servicios web).
- **HTTPS** en el puerto 8443 también es válido (es el alternativo más común a 443).

¿Por qué se usan diferentes puertos?

- HTTP y HTTPS son **protocolos diferentes** (HTTPS incluye cifrado TLS/SSL).
- Usar diferentes puertos:
 - Facilita la configuración y pruebas locales sin colisiones.
 - Permite dejar puertos 80 y 443 libres para otros servicios públicos si es necesario.
 - Aumenta el control sobre reglas de firewall o balanceadores de carga.
 - Separar puertos evita confusión, facilita debugging, y cumple mejor con estándares y documentación futura.
- ***Si se desea forzar la conexión a HTTPS, se tiene que hacer un direccionamiento del puerto 83 al 8443, esto se realizaría desde el proxy.***

Escenario recomendado:

Protocolo	Puerto	Justificación
HTTP	83	Opcional (si quieres seguir ofreciendo HTTP)
HTTPS	8443	Totalmente separado para tráfico cifrado

CONCLUSIÓN

Como parte del análisis de seguridad, se ha verificado que las conexiones realizadas tanto mediante el protocolo **HTTPS (puerto 8443) como por medio de net.tcp (puerto 84) viajan cifradas de extremo a extremo.**

En el caso de HTTPS, se constató que el certificado SSL utilizado es válido, emitido por una entidad certificadora reconocida y correctamente instalado en el servidor. Se recomienda mantener habilitada únicamente la conexión segura por HTTPS y redirigir cualquier intento de conexión por HTTP (puerto 83) hacia HTTPS, a fin de prevenir el uso involuntario de canales no cifrados.

Asimismo, se valida que el canal net.tcp cuenta con configuración de seguridad en modo **Transport**, lo cual garantiza la confidencialidad e integridad de los mensajes.