

Curso de Interoperabilidad entre Sistemas de Información en Salud

Tarea 3

Objetivo: probar la comunicación con el PACS y servicios DICOM.

Planteo: crear un cliente MLLP para DCM4CHEE y probar comandos de DCM4CHE Toolkit.

Entrega: ¡leer atentamente!

1. La entrega debe incluir el código fuente, el binario compilado, documentación de la solución propuesta, problemas que encontró y cómo los solucionó.
2. La documentación debe incluir las instrucciones exactas para compilar y ejecutar él o los programas o scripts entregados. Si no se incluyen, **no se podrá evaluar la tarea.**
3. Si utiliza librerías o código de otros proyectos, debe agregar una referencia a qué proyectos y qué versiones de los mismos está utilizando. Siempre debe acreditar el trabajo de otros colegas.
4. Seguir la estructura de proyecto en carpetas:
 - a. /src < código fuente
 - b. /lib < librerías externas y dependencias
 - c. /bin < binarios compilados
5. **Entregar proyecto en archivo ZIP**
 - a. **Nombre: "nombre_apellido_tarea_3.zip"**
6. Se sugiere utilizar Java o Groovy para el desarrollo. Se aceptarán entregas en otros lenguajes de programación mientras incluyan instrucciones para correr las soluciones, incluyendo construcción del entorno y dependencias de librerías externas.
7. Incluir las capturas de paquetes de WireShark* en formato **pcapng**.
 - a. pcapng es un formato de texto que permite expresar información de tráfico en redes: <https://www.winpcap.org/ntar/draft/PCAP-DumpFileFormat.html>

Plazo: ver en campus virtual de ACHISA

Puntaje máximo: 30 / 100

* Si tiene problemas con la captura de paquetes locales en máquinas Windows, tiene dos opciones: 1. crear una máquina virtual y comunicar el host con la virtual, 2. probar con RawCap <http://www.netresec.com/?page=RawCap> (RawCap genera archivos pcap que pueden abrirse con Wireshark para analizar). Ver: <http://carminedimascio.com/2014/03/rawcap-and-wireshark-how-to-capture-and-analyze-local-traffic-from-host-machine-to-itself/>

Planteo:

Para poder realizar esta tarea necesita instalar DCM4CHEE. La información al respecto está en el Campus Virtual.

Parte 1: envío de mensajes HL7 al PACS

1. Programar un cliente MLLP para el envío de mensajes:
 - a. HL7 v2.3.1 ORM (para agregar items a la worklist del PACS) y
 - b. HL7 v2.3.1 ORU (agregar reportes estructurados (SR) al PACS).

Referencia: <https://dcm4che.atlassian.net/wiki/display/ee2/HL7+Server>

Nota: puede utilizar HAPI para crear el cliente MLLP.

Parte 2: probar comandos del DCM4CHE Toolkit contra el PACS

Debe documentar: cada comando ejecutado, los resultados obtenidos, y explicar lo que obtuvo (como lo explicamos en la clase 3).

1. Probar la consulta de Modality Worklist usando DCM4CHE Toolkit (DCM4CHE2), usar distintos criterios y describir los resultados obtenidos. Se obtendrán datos de las transacciones HL7 ORM realizadas en el punto 1.
 - c. Comando dcmmwl: <https://dcm4che.atlassian.net/wiki/display/d2/dcmmwl>
2. Cargar estudios de DICOM para poder probar las consultas al PACS (Query/Retrieve)
 - d. Comando dcmsnd: <https://dcm4che.atlassian.net/wiki/display/d2/dcmsnd>
3. Probar el servicio Query/Retrieve, usando distintos criterios en los 4 niveles (paciente, estudio, serie e imagen). Describir los resultados obtenidos para cada nivel y distintos criterios:
 - e. Comando dcmqr: <https://dcm4che.atlassian.net/wiki/display/d2/dcmqr>
 - f. A nivel de paciente, usar como criterio* (entrada) de la búsqueda el nombre del paciente, identificador, fecha de nacimiento, sexo.
 - i. Puede hacer una sola búsqueda con los 4 parámetros o comenzar con 1 parámetro e ir sumando otros, o sea 4 búsquedas.
 - g. A nivel de estudio, usar como criterio* (entrada) de la búsqueda el identificador de estudio (Study Instance UID), fecha del estudio, y rango de fechas.
 - i. Hacer 3 búsquedas, una por cada parámetro.
 - h. A nivel de series, usar como criterio* (entrada) de la búsqueda el identificador de estudio (Study Instance UID).
 - i. Hacer 2 búsquedas, una por cada parámetro.
 - i. A nivel de imagen, usar como criterio* (entrada) de la búsqueda el identificador del estudio (Study Instance UID), identificador de la serie (Series Instance UID).
 - i. Hacer 2 búsquedas, una por cada parámetro.

* Usar como "criterio" de búsqueda, significa que cuando se realiza la búsqueda, se le da un valor al parámetro de entrada, no que se está pidiendo ese valor como resultado, ejemplo:

Esta consulta especifica el parámetro de entrada (criterio de la búsqueda):

```
dcmqr DCM4CHEE@192.168.1.111:11112 -qPatientName=MISTER -P
```

Esta consulta especifica los parámetros de salida (resultado):

```
dcmqr DCM4CHEE@192.168.1.132:11112 -P -r PatientID -r PatientName -r PatientSex -r PatientBirthDate
```

Parte 3: DICOM Broker (opcional)

4. Probar DICOM Broker para consultas al PACS y envíos de informes.
 - j. Monitorear con Wireshark para hacer ingeniería inversa: debe indicar cuándo se realizan las consultas por Query/Retrieve, cuándo se hacen pedidos WADO y cuándo se envían mensajes ORU por MLLP hacia el PACS.
 - k. Este análisis debe estar fundamentado por las capturas de Wireshark, las cuales deberá describir en la documentación entregada.

Para conectar DICOM Broker al DCM4CHEE, debe cambiar la configuración de conexión antes de correr DICOM Broker:

- Ver: <https://github.com/ppazos/cabolabs-dicombroker/blob/master/grails-app/conf/BootStrap.groovy#L77-L88>
- Modificar remoteIP con la IP de donde está instalado el PACS (127.0.0.1 si está instalada en la misma máquina donde corre DICOM Broker)
- Modificar remotePort si se cambió el puerto por defecto del DCM4CHEE, por defecto es 11112.

Se pide:

1. Monitorear todas las comunicaciones con Wireshark* y describir los resultados de todos los envíos de mensajes al PACS.
 - a. Se debe entregar la captura de paquetes de Wireshark en formato **pcapng**.
 - b. Se recomienda agregar una captura de pantalla de los paquetes capturados por Wireshark y una breve descripción de las comunicaciones (protocolos, direcciones IP, puertos, tipos de mensajes intercambiados, interpretación del contenido de los mensajes, etc).
2. Para las consultas con **dcmwl** y **dcmqr**, se pide describir parámetros usados para ejecutar los comandos, y los resultados que obtiene.

* Si tiene problemas con la captura de paquetes locales en máquinas Windows, tiene dos opciones: 1. crear una máquina virtual y comunicar el host con la virtual, 2. probar con RawCap <http://www.netresec.com/?page=RawCap> (RawCap genera archivos pcap que pueden abrirse con Wireshark para analizar). Ver: <http://carminedimascio.com/2014/03/rawcap-and-wireshark-how-to-capture-and-analyze-local-traffic-from-host-machine-to-itself/>