Application Layer

Transcribed on July 14, 2025 at 3:23 PM by Minutes AI

Speaker 1 (00:03)

Hola a todos y bienvenidos a esta sesión del módulo 3 donde vamos a hacer una introducción a la capa de aplicación.

Soy Javier Álvarez y voy a ser vuestro profesor en esta sesión.

A lo largo de esta sesión veremos qué es la capa de aplicación, qué compone la capa de aplicación en el modelo TCPIP y en el modelo OSI y también hablaremos de los protocolos que componen esta capa de aplicación.

En cuanto a qué es esta capa de aplicación es la última de las capas tanto del modelo OSI como del modelo TCP IP y ofrece a las aplicaciones, ya sean del usuario o no, la posibilidad de acceder a los servicios de las demás capas y también define los protocolos que utilizan las aplicaciones para intercambiar datos, como por ejemplo el correo electrónico que aquí utiliza el protocolo SMTP, también gestores de base de datos o protocolos de transferencia de ficheros como el protocolo FTP, entre otros.

El usuario normalmente no interactúa de forma directa con el nivel de aplicación, es decir, con esta capa lo que suele hacer es interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente.

Por ejemplo, un usuario no manda una petición GET para obtener el index HTML de una página, lo que hace es interactuar con su navegador para, introduciendo la URL, obtener esto de forma subyacente.

Aquí tenemos de nuevo ambos modelos, a la izquierda el modelo TCP IP y a la derecha el modelo OSI.

La capa de aplicación representa en el modelo TCP IP un único estrato o nivel, a diferencia del modelo OSI donde la capa de aplicación se divide en sesión, presentación y aplicación.

Esto es así porque en el modelo OSI la capa de aplicación se limita estrictamente a proporcionar la interfaz entre el software de aplicación y las capas inferiores del modelo, como las capas de presentación y de sesión.

Las capas de presentación y sesión a su vez se encargan de tareas específicas como la representación de los datos y el mantenimiento de la comunicación entre aplicaciones respectivamente.

These notes were taken with Minutes AI (https://myminutes.ai)

En el modelo TCP IP estas funciones se combinan en una sola capa llamada capa de aplicación.

A lo largo de estas sesiones hemos ido hablando de tramas en la capa de enlace, paquetes o datagramas en la capa de Internet y de segmentos en la capa de transporte.

Ahora en la capa de aplicación hablamos principalmente de datos en términos de mensajes que posteriormente se irán encapsulando.

En esta capa de aplicación encontramos multitud de protocolos.

Tenemos el protocolo de transferencia hipertexto, que es un protocolo utilizado para transferir datos en la web.

También tenemos el sistema de nombre de dominios, un sistema que lo que hace es traducir los nombres de dominios legibles por humanos, como por ejemplo Sample Run, en direcciones IP numéricas.

El protocolo de transferencia de ficheros o protocolo FTP, utilizado para transferir archivos entre un cliente y un servidor en una red.

Secure Cell Protocol o SSH, protocolo de red que nos permite a los usuarios conectarnos de forma segura a un servidor remoto a través de una conexión cifrada.

Y también otro de los que menciono por aquí, el Sample Mail Transfer Protocol o SMTP, que es un protocolo de comunicación utilizado para enviar correos electrónicos a través de Internet.

SMTP es el estándar para el envío de mensajes de correo entre servidores de correo electrónico.

Y sin más, llegamos al final de esta clase.

Hola a todos y bienvenidos a esta sesión del módulo 3, donde vamos a hablar de los protocolos de la capa Aplicación.

En esta clase vamos a ver con qué entorno de laboratorio vamos a estar trabajando el protocolo de transferencia hipertexto HTTP, el sistema de nombres de dominios DNS y el protocolo de transferencia de ficheros FTP.

El entorno de laboratorio con el que vamos a estar trabajando va a constar de una máquina Kali y también de una máquina Ubuntu Server 1.

En esta máquina, el Ubuntu Server 1, lo que vamos a hacer es setear el servidor FTP para posteriormente hacer y realizar diferentes llamadas desde la máquina Kali.

Por tanto, vamos a comenzar hablando del protocolo HTTP.

These notes were taken with Minutes AI (https://myminutes.ai)

Es un protocolo de transferencia de hipertextos que lo que se usa es para acceder a los datos de la web.

Utiliza por debajo el protocolo de transporte TCP.

El cliente inicia la conexión TCP al servidor, el servidor la acepta, se transfieren los mensajes, los datos y final, la conexión TCP se cierra.

Se dice que HTTP es un protocolo sin estado, ya que no guarda ninguna información respecto a peticiones anteriores.

Es por eso que se reconocen dos formas o dos estados de HTTP no persistente y HTTP persistente.

La versión no persistente de HTTP consiste en que se envía un único objeto en cada conexión TCP.

Como veis a la izquierda, si queremos enviar tres objetos se va a tener que establecer la conexión y cerrar durante tres veces.

Esto ocurría en las primeras versiones de la versión la 1, la 1.1, etc.

Luego tenemos la versión de HTTP persistente.

Esto se extendió sobre todo en la versión 2.

Ahí se comenzó a utilizar de esta forma y se establece la conexión una única vez y se aprovecha dicha conexión para enviar diferentes objetos y finalmente se cierra.

En la versión 3 de HTTP, que salió en 2022, se dejó de usar TCP, por tanto ya no estamos hablando de persistencia ni no persistencia.

Aquí directamente se utiliza otro protocolo, uno creado por Google llamado Twitch y con UDP.

Como podéis ver en esta tabla.

HTTP además tiene diferentes métodos de peticiones.

Tenemos el método READ, el POS, el PUT y el DELETE.

Estos son los más conocidos.

Como veis en esta imagen podéis detenerla y podéis ver los diferentes RFC que están relacionados con cada método.

También podéis ver si en esta petición de, por ejemplo, GET, normalmente se lleva BODY en la request o en la response la respuesta o no.

These notes were taken with Minutes AI (https://myminutes.ai)

Por ejemplo en HEAD, en la petición HEAD, el BODY en la REQUEST es opcional, pero en la respuesta nunca lo puedes llevar.

Ahora vamos a ver un ejemplo de una solicitud HTTP y su respectiva respuesta.

Como podéis ver, en la primera línea vemos el método que es de tipo GET y además vemos que se está haciendo una solicitud al recurso raíz, de ahí la barra, y también que se está utilizando la versión 1.1 de HTTP.

El resto son las diferentes cabeceras que lleva esta solicitud HTTP.

Nosotros nos vamos a centrar en una, en la cabecera HOST.

Aquí le estamos indicando que el nombre de dominio del servidor al que se está solicitando es en este caso.

Por último tendremos una línea en blanco y si el input llevara algún tipo de BODY se indicaría después de esa línea en blanco.

Allí tenemos la respuesta.

La primera línea, como podéis ver, es la versión de HTTP y nos indica el código de estado 200 OK.

Cuando vemos un código de estado que empieza por 2, significa que la petición ha ido de forma correcta, satisfactoria.

Si empezara por 1, como podéis ver a la derecha, nos dice algo informacional.

Si empieza por 3 se trata de una redirección y si empieza por 4 es un error del cliente.

Cuando ese código de estado empieza por 5, se trata de un error del servidor.

De esa forma podríamos saber si es que el servidor está fallando o nosotros no estamos enviando la petición de forma correcta.

Al igual que antes, tenemos diferentes etiquetas, diferentes headers, y me voy a centrar en una en la etiqueta ETAG.

Esta etiqueta se llama etiqueta de entidad única para el recurso solicitado y al final lo que nos indica es si el servidor, si la respuesta ha cambiado respecto a la que está cacheada, de esa forma se actualiza o no.

Por último tenemos la línea en blanco, como en todas las peticiones y respuestas de tipo HTTP y después el fragmento del cuerpo de la respuesta, que en este caso es un documento HTML muy simple que muestra al final un help.

Ahora vamos a ver un servidor HTTP construido en Flash y realizando algunas peticiones desde Kali.

These notes were taken with Minutes AI (https://myminutes.ai)

Vale, pues ya estamos por aquí en nuestra máquina Kali.

Fijaros aquí en el desktop tengo este server py, un fichero de python from flash donde he ejecutado y he levantado diferentes endpoints de un servidor HTTP.

Estos endpoints son add con el método tipo POST, update con el método put y get con el método get to.

Todo ello para probar las diferentes peticiones HTTP que vamos a realizar mediante la línea de comandos utilizando Google.

Por tanto vamos a levantar este servidor HTTP y para ello necesitamos instalar flash.

Escribimos por aquí ppistol flash y una vez hecho esto podemos escribir ya python server py y fijaros que ya está el servidor ejecutándose en un entorno de desarrollo, pero se está ejecutando en el puerto 5000.

Vamos a abrir otra terminal por aquí, voy a volver a hacer zoom y lo que vamos a ejecutar por aquí es el siguiente comando.

Ya tengo por aquí el comando escrito.

Vamos a ejecutar un comando de tipo POST con la cabecera content type application JSON y el body key value john do Todo esto lo vamos a redirigir a la petición localhost a.

Damos a enter y vamos a ver por aquí que nos dice que los datos se han añadido de forma exitosa.

Por tanto ahora si hacemos una petición muy simple, la siguiente.

Voy a borrar todo esto y vamos a hacer una petición de tipo GET.

Se da por hecho que va a ser de tipo GET si no se indica nada y vamos a poner por aquí get y, que es el key del dato que acabamos de añadir.

Y por aquí nos va a devolver Yondou.

¿Qué vamos a hacer ahora?

Vamos a modificar ese nombre, en lugar de yondo vamos a poner Jane Doe, por ejemplo.

Entonces aquí lo que tenemos que cambiar es que en lugar de post va a ser put en el body.

Ahora tenemos que eliminar la key, porque la key se la vamos a indicar de otra forma.

Aquí le vamos a indicar el nuevo nombre, que va a ser por ejemplo Jane Doe, así Y aquí le tenemos que indicar el update porque es el endpoint ash que queremos llamar y aparte la key que queremos modificar.

These notes were taken with Minutes AI (https://myminutes.ai)

Damos a enter y nos dice que el data ha sido actualizado de forma exitosa por Por último vamos a obtener estos datos que acabamos de actualizar con la petición anterior que hacíamos localhost 5000 get damos a enter y ahí tenemos John Doe y Jane Doe.

Por tanto ya hemos modificado, añadido y obtenido los diferentes datos de este servidor HTTP.

Pasamos a ver el sistema de nombres de dominio.

Las personas preferimos utilizar nombres en lugar de direcciones IP, porque al final es algo más fácil, y para esto necesitamos un sistema que sea capaz de traducir esos nombres de dominio en direcciones IP.

Los sistemas de nombre de dominio son bases de datos, podríamos decirlas así, que además están distribuidas e implementadas en jerarquías de muchos servidores de nombre.

Y os preguntaréis ¿Por qué una jerarquía y muchas bases de datos y no un sistema DNS centralizado en un único sitio?

Pues la verdad que esto no es lo correcto, ya que provocaría dos que haya un único punto de fallo y y además que al existir un único servidor de nombre de dominios hay un volumen muy grande de tráfico y por ende enormes latencias.

En la imagen que veis en pantalla podemos ver como cuando un usuario escribe un nombre de dominio en su navegador web, por ejemplo WWW.

WIKIPEDIA.

ORG, el navegador lo primero que envía es una consulta al servidor DNS raíz.

Este servidor es el primer punto de contacto en la jerarquía DNS y contiene referencias a los diferentes servidores de nivel superior, los servidores TLD, para cada dominio.

Es decir, tenemos un servidor para COM, para.

ORG, para NET, para ES, etc.

En este caso el servidor DNS raíz no responde a la consulta con la dirección IP del servidor DNS TLD.

Este servidor a su vez le vamos a preguntar por el nombre WIKIPEDIA.

ORG y nos va a responder con el DNS o el servidor DNS autoritario para el registro WIKIPEDIA.

ORG y finalmente cuando peticionemos a ese servidor autoritario ya nos va a responder con la dirección IP del servidor WIKIPEDIA.

These notes were taken with Minutes AI (https://myminutes.ai)

ORG o WWW WIKIPEDIA.

ORG.

Por tanto, fijaos que ha sido una jerarquía, hemos preguntado primero al raíz, luego al de nivel superior y finalmente al autoritario.

Tras esto, lo que hace nuestro navegador es almacenar este registro, esta respuesta en caché para futuras consultas y así acelerar el acceso a los sitios que se han visitado anteriormente.

Por aquí he puesto un ejemplo, lo he creado yo a mano, de un registro DNS.

Fijaros que tenemos registros de tipo nameserver.

Estos registros nos indican a qué se servidores de nombre dominio, podemos peticionar para resolver el dominio en cuestión, es decir, son servidores de nombre autorizados.

Luego tenemos registros de tipo A al final, este registro se utiliza para asociar el nombre de dominio con una dirección IPV específica.

El registro A es lo mismo que el A pero para IPV.

Registramos y asociamos un nombre de dominio o subdominio a una dirección IPV específica.

También tenemos registros de tipo CNAME, registro que se utiliza para crear un alias de nombre dominio.

Por ejemplo, tenemos un servidor web con el nombre WWW y queremos que también responda a sample, por tanto podemos crear un registro CNAME que apunte sample chrome a sample chrome.

También tenemos el registro tipo NX, este es para correo y especifica el servidor de correo responsable en el que recibir los correos electrónicos destinados a un dominio específico, por ejemplo infol, pues aquí tendríamos que tener un registro para dicho dominio.

Y por último tenemos el registro TXT.

Este registro se utiliza para almacenar texto arbitrario, normalmente asociado a políticas de seguridad.

Por ejemplo, se utilizan a menudo para la validación del dominio y la configuración de seguridad, como SPF Sender Policy Framework y Domain Keys Identified Mail.

Pasamos a ver el último protocolo Protocolo de transferencia de ficheros, un estándar de utilizado para transferir archivos entre un cliente y un servidor en una red.

These notes were taken with Minutes AI (https://myminutes.ai)

Utiliza la capa de transporte TCP y también el protocolo IP.

Los casos de uso más relevantes de este El desarrollo de sitios web, utilizado por ejemplo para subir los archivos de tipo HTML, CSS, JavaScript e imágenes a un servidor.

También utilizado para compartir archivos.

También utilizado para copias de seguridad y almacenamiento.

Es decir, utilizamos FTP para subir estas copias de seguridad a servidores remotos donde la almacenamos.

Y también es utilizado, o más bien era utilizado antes, para distribuir software y medios.

Antes las compañías cuando querían hacer público un nuevo software o una actualización, lo hacían a través de sus servidores FTP.

Tenemos varios clientes de tipo FTP.

Tenemos filesilla, tenemos cyberduck, estos son los más relevantes, ambos de código abierto y gratuitos.

Y por aquí también tenemos los comandos más populares de FTP.

Pueden variar según la implementación específica del servidor, pero hay algunos que son los más comunes, que son estar disponibles en la mayoría.

Tenemos como podéis ver LS, CD, PWE, MTADIR, R, MEDIR y CHMOD, pues son comandos que llamemos en Linux a los que estamos acostumbrados.

Sí que hay diferencias, por ejemplo get y put para descargar y subir respectivamente un archivo al cliente o del cliente.

También tenemos delete para eliminar un fichero, rename para como podéis imaginar, renombrar ese fichero y por último tenemos quiz para cerrar la conexión, FTP o help para mostrar ayuda de los comandos disponibles en esa versión específica de FTP.

Ahora vamos a ir al entorno de laboratorio donde vamos a setear un servidor FTP en la máquina Ubuntu Server y vamos a utilizar la máquina Kali como cliente para subir y descargar algunos ficheros.

Ya estamos aquí en nuestro laboratorio y lo que vamos a hacer ahora es instalar un servidor FTP en la máquina Ubuntu Server y para ello vamos a ir por aquí a esta máquina y vamos a escribir sudo apt app sudo apt install vsftpd que va a ser el servidor de nombre, el daemon que vamos a estar instalando y ejecutando.

Voy a poner i para que se instale automáticamente y nos vemos en breve.

These notes were taken with Minutes AI (https://myminutes.ai)

Una vez instalado lo que podemos hacer es modificar.

Lo que podemos hacer es modificar el fichero de configuración de este servidor ftpd, para ello vamos a etctpd conf y lo modificamos y aquí lo que debemos hacer es descomentar esta configuración que dice Write enable yes.

Guardamos por aquí y vamos a reiniciar este servidor ftpd, para ello escribimos sudo systemctl restart vsftpd de nuevo.

Ahora si con este comando lo modificamos y en lugar de restar hacemos un status, vamos a poder ver que está activo y corriendo.

Pues una vez hecho esto vamos a escribir por aquí FTP.

Nos conectamos a este servidor FTP, nos pide el usuario y la contraseña y ya estamos dentro.

¿Qué vamos a hacer ahora?

Pues podemos escribir el comando help para ver todos los comandos que se permiten en este servidor FTP específico.

¿Qué vamos a hacer ahora?

Pues nos podemos descargar un fichero de la máquina remota, para ello vamos a hacer un ls y vemos que está el fichero HELLO txt, pues podemos descargarlo gracias al comando get, ponemos gethellow txt y nos los descarga en nuestro local.

Fijaros que si yo aquí me abro otra terminal y hago un cad HELU txt puedo ver el fichero que dice hello from server 1.

También por supuesto puedo subir ficheros a la máquina remota, para eso hemos habilitado la escritura, entonces podría escribir putserver py o server pi para decirle que está en este directorio y fijaros que lo sube a la máquina remota y ahora si hacemos un ls ya tenemos ahí ese fichero server py en dicha máquina.

Y con esto llegamos al final de esta sesión donde hemos visto con un poco más de detalle los protocolos como HTTP, el protocolo FTP y también el sistema de nombres de dominio DNS.

Sin más me despido y nos vemos en la próxima sesión.

These notes were taken with Minutes AI (https://myminutes.ai)