Informe de la cuarta actividad GSI

@ALVPI

Contents

1 Tarea 1: ¿Qué sentido tiene los puertos a nivel de transporte que tiene la UVa?

Hagamos un nmap para ver el estado de los puertos de la página web de la UVa.

```
~/Desktop git:(master)±112 (1m 30.54s)
nmap -Pn uva.es

Starting Nmap 7.95 ( https://nmap.org ) at 2024-12-12 19:36 CET
Nmap scan report for uva.es (157.88.25.8)
Host is up (0.0030s latency).
rDNS record for 157.88.25.8: www.uva.es
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE  SERVICE
20/tcp    closed ftp-data
80/tcp    open   http
443/tcp   open   https
465/tcp   closed smtps
587/tcp   closed submission
993/tcp   closed imaps
2196/tcp  closed unknown
5222/tcp  closed xmpp-client
10000/tcp closed snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 90.45 seconds

~/Desktop git:(master)±112
```

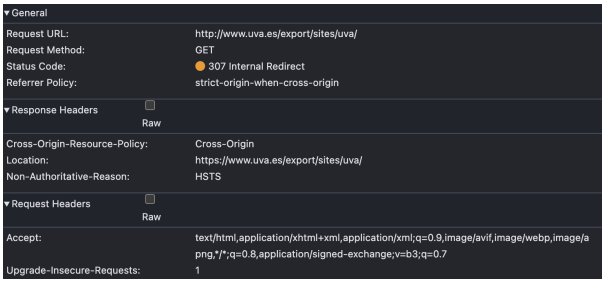
Figure 1: Estado de los puertos en UVa.es

Abrimos un navegador (Google Chrome en mi caso) y presionamos **f12**, lo que nos abriría las *DevTools* que son un conjunto de herramientas integradas en el navegador que permiten a los desarrolladores web realizar tareas de depuración, análisis y optimización de aplicaciones web, seleccionamos el apartado de *Network*, la cual es una sección que nos permite monitorear y analizar las solicitudes de red, incluyendo tiempos de carga, encabezados y respuestas;posteriormente vamos a la
Ahora intentamos acceder a la web de la UVA utilizando http en vez https (protocolo por defecto) y conseguimos acceder a la web de la UVa pero al ir a la URL, aunque haya escrito http, pone https, es decir, ha cambiado automáticamente de un protocolo a otro.

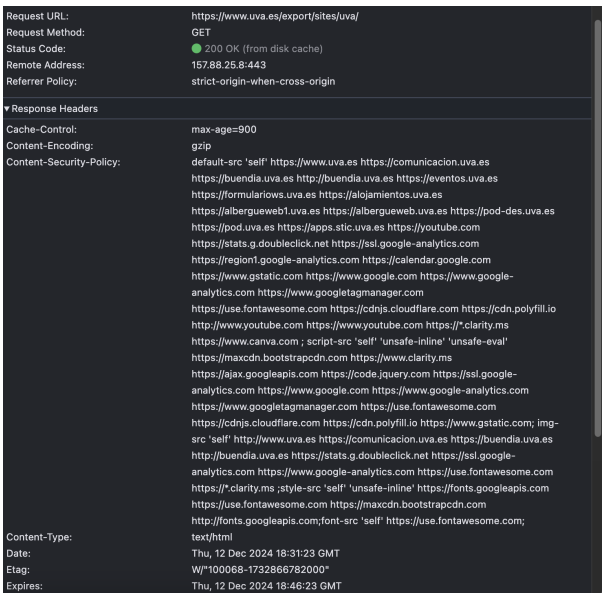
uva/	307	document / Re...	Other	0 B	1 ms
uva/	200	document	/export/sites/uva/	(disk cache)	3 ms

Figure 2: Datagrama con dos paquetes de tipo documento

1.1 Requests Methods



(a) Petición de redirección.



(b) Petición GET HTTPS.

Figure 3: Las dos primeras primitivas de la comunicación con la página web de la UVA

Si nos fijamos en la imagen, vemos que esta petición lo que está haciendo es automáticamente cambiar http por https, lo cual se hace de manera automática.

1.2 Por qué se realizan el cambio automático de HTTP a HTTPs

Está pregunta se puede abordar desde diversos puntos.

1. **Facilidad para los usuarios**: Muchos de los usuarios de internet al buscar una página web, **no** especifican el protocolo, simplemente escriben una url en el navegador, esto hace que por defecto se establezca una comunicación http, la cual va al puerto 80, si este no está habilitado, redireccionamos directamente al puerto 443 (https) para poder acceder a la web.
   Además, tenemos que tener en cuenta, no todas las personas que utilizamos internet tenemos el interés o el conocimiento para tener nuestros dispositivos actualizados ni el software del mismo, pero eso no ha de ser una condición restrictiva para el uso de la red de redes por ende, al hacer este redireccionamiento permitimos a estos usuarios seguir utilizando internet sin esa preocupación.

2. **Seguridad**: Al no especificar un protocolo y que por defecto se utiliza http supone un riesgo para los usuarios puesto que la comunicación entre los mismos y el servidor **no** va cifrada, sino que se transmite por texto plano, lo cual implica un riesgo para los usuarios y los posibles datos que intercambie con el servidor.

3. **SEO**: Los buscadores más utilizados, basan los resultados de nuestras búsquedas en función de diversos factores sobre las páginas webs, por ende, como también aparecen páginas que no son seguras, puesto que utilizan `http` para mejorar la experiencia del usuario se redirecciona a https.

4. **Evitar ataques de downgrade a las páginas webs**: Este tipo de vulnerabilidades, se basan en intentar acceder a una web, empleando un protocolo mas antiguo o inseguro para poder explotar posibles vulnerabilidades sobre los mismos.

5. **Uso de ciertos certificados digitales**:Algunos certificados digitales como puede ser `Let's Encrypt`, asignan los certificados de manera dinámica a los propietarios del dominio, de tal manera que emplea el **DV** ("*Domain Validation*") para verificar si eres tú el propietario de dicho dominio empleando el *http-01 challenge* el cual envía un token al al cliente que solicita el certificado y este tiene que crear un archivo público mediante http, si *Let's Encrypt* puede acceder confirma que el solicitante tiene control sobre el dominio.

6. **Reducción de la congestión del tráfico de red**: Las peticiones de tipo http son más livianas puesto que no tienen que securizar la información. Por lo que gracias a aceptar este tipo de peticiones, podemos gestionar la carga de los servidores, en situaciones de alto tráfico, puesto que procesamos la solicitud y posteriormente utilizamos la pasarela para utilizar https.

7. **Rastreo y monitorizacón**: Los administradores de la red/web, pueden utilizar los logs de peticiones para establecer patrones de comportamiento sobre las peticiones que recibe la aplicación.

8. **Proxies/Proxies inversos**: Podemos gestionar el acceso a la página web de dos maneras, para el frontent podemos emplear peticiones http, puesto que su carga es más liviana y en caso de que el cliente deba acceder al bakends, es cuando empleamos https para securizar dicha comunicación sin sobrecargar el sistema y mejorando la compatibilidad con el sistema.

2 Tarea 2 :Explica con un diagrama de secuencia por qué en un certificado digital se usa criptografía de clave pública y privada?

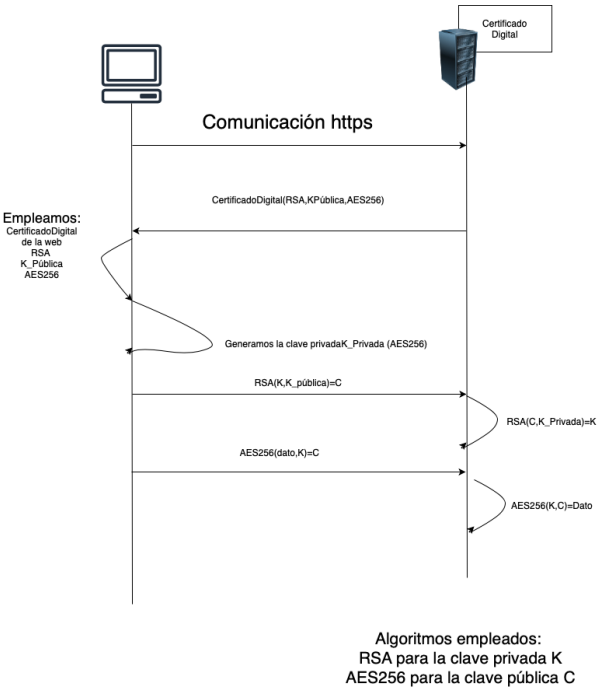


Figure 4: Funcionamiento de clave pública-privada en cliente-servidor

Figure 5: Cifrado clave pública-privada policía nacional

Comenzaremos entendiendo que son estos dos algoritmos:

1. **RSA** (”*Rivest-Shamir-Addleman*”): Es un cifrado asimétrico de clave pública y privada con la ventaja de que no es necesario compartir previamente la clave privada.

   *Funcionamiento: Usa dos claves: una clave pública para cifrar datos y una clave privada para descifrarlos.

   *Su seguridad se basa en la dificultad de factorizar números grandes (matemáticas de números primos).

2. **AES-256**(”*Advanced Encryption Standard*”) es un cifrado simétrico.

   *Funcionamiento: Empleamos la misma clave para cifrar y descifrar los datos (de 128 bits), empleando para la clave, bloques de 256 bits, lo que unido a sus operaciones de intercambio y permutaciones de bits, lo convierten en un sistema bastante robusto.

Una vez que ya tenemos claro los mecanismos de cifrado que hemos empleado en el ejemplo, vamos a proceder a explicar el proceso empleando conexiones https.

RSA (empleo de cifrado simétrico):

1.El cliente solicita conectarse con el servidor, este envía su clave pública al cliente como parte del certificado de seguridad.
Dicho certificado ha de estar firmado por una **CA**(”*Certification Authority*”) la cual es la que aporta veracidad sobre el mismo.

2. El cliente comprueba que la clave pública es válida y que corresponde con quién dice ser el host, en caso de no pasar esta verificación, se nos muestra un mensaje sobre la inseguridad del mismo.

3. Una vez que ya se ha generado esa ”seguridad”, el cliente cifra el dato con la clave pública del servidor, generando así un **secreto compartido**, que solo podrá ser descifrado mediante la clave privada de servidor.

4. El servidor descifra el mensaje con su clave privada, para poder tener acceso también a ese **secreto compartido**.

1. **AES(empleo de cifrado simétrico))**

5.El secreto será empleado a partir de ahora para asegurar que la comunicación es segura,