Tarea 1: Desarrollo Seguro y gestión de identidades

```
public int fibonacci(int n){
    if (n == 0 || n == 1){
        return 1;
    }else{
        return fibonacci(n - 2) + fibonacci(n - 1);
    }
}
```

- *¿Es correcta la implementación del código anterior para el cálculo de la sucesión de fibonacci?*

  La implementación es correcta. En este caso se ha decidido usar la recursividad en lugar de bucles como for, por ejemplo.

- *¿Exista alguna limitación intrínseca a la codificación realizada?*

  La limitación de este código viene en el rendimiento. Usar funciones recursivas posiblemente nos genera un código mas elegante, pero son más lentas (en el ejemplo en llamadas a números altos) ya que se llaman a si mismas constantemente y multitud de veces.

Verificación y pruebas:

- *¿Para qué sirven cada una de las siguientes herramientas?*

MetaSploit:

Es el nombre del proyecto Open Source utilizado en el trabajo de auditoria informática proporcionando información sobre vulnerabilidades y analizando los procesos de pentesting.

Dentro del proyecto tenemos el framework mas conocido, también llamado Metasploit (framework) El cual es un conjunto de herramientas desde donde se pueden desarrollar y ejecutar exploits. También aporta otras herramientas para buscar vulnerabilidades, información, etc

Acunetix:

Es un escáner web de vulnerabilidades que funciona de forma automática. Permite automatizar tareas, ahorrando mucho tiempo a las personas que auditan los sistemas.

Nessus:

Escáner muy flexible con funcionalidades como: descubrimiento activo de redes, escaneo de vulnerabilidades, políticas de auditorías, etc

BeeF:

“The Browser Explotation Fraework” Herramienta de test de penetración centrada en el navegador web. Permite evaluar la seguridad desde el lado del cliente en los navegadores web.

Wireshark:

Nos permite analizar los protocolos de red y ver lo que esta sucediendo en ese nivel. Incluye herramientas que permiten capturar análisis de red, visualizar los paquetes que tenemos por nuestra red, etc

Shodan:

Es un motor de búsqueda de fuentes abiertas el cual nos permite encontrar información en toda la red de diversas maneras: a través de protocolos, IPs, metadatos etc

Por ejemplo se puede utilizar para ver que información está exponiendo una compañía en internet y ver si hay algo que potencialmente peligroso para la seguridad de la misma.

- *¿Qué otras herramientas similares se podrían usar para realizar un test de penetración?*

La distribución de Kali Linux es un buen ejemplo de SO que nos permite hacer test de penetración, ya que dispone de muchas herramientas: Nmap, Burp Suite, Nikto, Maltego, SQLMap, WhatWeb, Hydra, el propio MetaSploit, wireshark, etc

Modelado de Amenazas:

- *¿SE DEBERÍAN CONSIDERAR LAS LIBRERÍAS DE CÓDIGO DE TERCEROS INCLUIDAS EN EL PROYECTO COMO DEPENDENCIAS EXTERNAS?*

  Si deberán ser incluidas como dependencias externas ya que no tenemos el control sobre esas librerías y podría ocurrir algún problema (como el reciente con log4j) en ellas, que repercutiese en nuestro sistema.

- *¿POR QUÉ SON DOS ACTIVOS DISTINTOS EL LOGIN DE USUARIO TIPO PROFESOR O ALUMNO (1.1) Y EL LOGIN DE LOS BIBLIOTECARIOS (1.2)?*

  Son activos u objetivos diferentes ya que para cada uno de los perfiles lo que podemos ver y hacer en la aplicación es diferente. Si un atacante consiguiese comprometer el login de un bibliotecario tendría ciertos accesos que como alumno no están disponibles.

- *Además del modelo STRIDE, existen otras clasificaciones de amenazas.*
  - *¿Qué otras clasificaciones alternativas existen?*
  - *¿Existe alguna preferencia entre STRIDE y alguna de ellas a la hora de elegir una categorización u otra en un modelado de amenazas?*
  - *¿Qué relación existe entre la categorización de amenazas y los CVSS?*

Hay mas modelos como: ASF (mencionado en los apuntes), PASTA, VAST, Trike, DREAD, OCTAVE, etc

Por lo que me he podido encontrar en internet el método por excelencia es STRIDE pero usado con CVSS, un método hibrido que modela las amenazas y las puntúa. Por lo tanto, los sistemas CVSS complementan a los de modelado de amenazas.

Del resto de modelos mencionados, algunos son mas ligeros, Como VAST, otros siguen una serie de pasos muy definidos como PASTA, etc en definitiva no hay una única solución para todos los problemas, pero es mas que recomendable realizarla para analizar las posibles amenazas y cuantificar su gravedad.

- *Como se mencionó, la lista de chequeos anterior aplica especialmente a entornos web, si estuviéramos en un escenario de aplicaciones móviles:*
  - *¿Qué otras validaciones adicionales se deberían aplicar?*

    Se me ocurren varias opciones:

    Validar como gestiona la aplicación los ficheros, en que lugares los guarda, como, con que permisos, etc

    Los permisos que la aplicación solicita para su funcionamiento.

    Como se puede interactuar con la aplicación: utilización de datos, sensores del móvil, etc

    Como es el proceso de login, si tiene 2FA, etc

  - *¿Y cuáles de las anteriores no tendrían sentido?*

    De las anteriores no tendrían sentido las relacionados con un entorno web: principalmente la gestión de cookies y de sesiones.

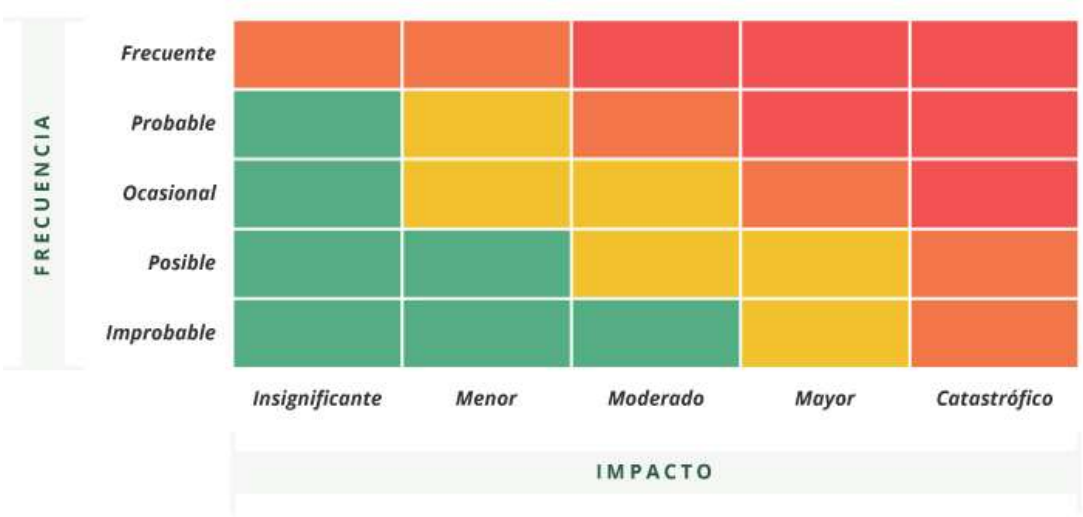
- *¿Qué es una matriz de riesgos?*

  Es una herramienta que permite evaluar de manera objetiva los riesgos de seguridad identificados que puedan aparecer.

- *¿Cómo se construye?*

  Se priorizan los riesgos identificados y estos se puntúan según la frecuencia con la que pueden aparecer y el impacto que pueden tener en el sistema.

  Estos datos se representan gráficamente:



- *¿Para qué es útil?*

  Ayuda a visualizar los riesgos inherentes en el proyecto y a su vez nos puede ayudar a tomar decisiones más rápidas sobre los mismos si estos aparecen.

Identidad Digital:

- *DE LOS TRES CANALES EXPUESTOS ANTERIORMENTE PARA PROVEER UN CÓDIGO DE UN SOLO USO (EMAIL, SMS O NOTIFICACIÓN A UNA APP)*
    - *¿Podemos considerar los 3 igual de seguros?*
    - *¿En caso negativo, cuál sería el orden de mayor a menor seguridad? ¿Por qué?*

No podemos considerar los 3 igual de seguros porque tanto él envió de sms, como por email puede verse afectado más fácilmente que el de una app por un uso mal intencionado. Por ejemplo, si el sistema se ve comprometido y se cambia el número de teléfono de destino del sms o email, implicando una suplantación de identidad.

Las apps de autenticación en 2 factores son más seguras, aunque dentro de estas, también hay unas mas seguras que otras. Por ejemplo, Google Authenticator no esta protegida por una contraseña y no se puede bloquear (siempre esta accesible), si perdemos/nos roban nuestro móvil y consiguen tener acceso al mismo, también lograrían entrar en esta app.

Así que los mas seguro que podemos usar es una app de autenticación en dos factores que utilice login como Authy o 1Password.

- *¿QUÉ MECANISMOS DE VERIFICACIÓN PODRÍAMOS UTILIZAR PARA VERIFICAR CORRECTAMENTE ESTOS OTROS PARÁMETROS ASOCIADOS A UNA IDENTIDAD?*
    - *Tarjeta de crédito/débito*
    - *Otra identidad digital del usuario (una red social por ejemplo)*

En el caso de las tarjetas de crédito se verifica la misma con el nombre del titular, número de la tarjeta, fecha de caducidad y CVV. Es la manera standard de verificar los datos de la tarjeta, pero para verificar que la misma pertenece a la persona que esta metiendo los datos normalmente se verifica esta identidad a través de una confirmación de una duración determinada a través de la app del banco correspondiente.

Vas a hacer un pago, introduces los datos de la tarjeta, esta se verifica y para verificar la identidad te envían una notificación push a tu app del banco, donde verificas la identidad y si todo es correcto puedes seguir con el pago.

También se esta empezando a utilizar la biometría, facial, voz, etc para verificar la identidad de la persona que es titular de la tarjeta. Las apps de los bancos están incorporando este tipo de servicios para verificar identidades.

En las redes sociales nos podemos encontrar un caso parecido al anterior, pero verificando la identidad a través de una app de 2FA o con confirmaciones de email o sms.

Hay una nueva manera de verificar identidades mediante la tecnología blockchain y los tokens no fungibles, conocidos como NFT. Es una manera de verificar que la persona que es propietaria de X token es la que esta intentando acceder a Twitter (por poner un ejemplo). Para ello esta usando su clave privada de la blockchain que se este usando para verificar que es el propietario de dicho token y por lo tanto quien dice ser.

- *¿QUÉ PARÁMETROS, CAMPOS O FLAGS INCORPORAN TODAS LAS COOKIES?*
- *¿PARA QUE SE UTILIZAN?*

Las cookies se utilizan para tres cosas: Gestión de sesiones de usuario, personalizaciones/configuraciones y para rastreo y análisis.

```
Set-Cookie: <name>=<value>[; <Max-Age>=<age>] [; expires=<date>][;
domain=<domain_name>] [; path=<some_path>][; secure][; HttpOnly]
```

Al generar una cookie,, Set-Cookie, le podemos pasar los siguientes parámetros:

Name: nombre de la cookie. Es el único parámetro obligatorio.

Max-Age: tiempo máximo de vida de la cookie.

Expires: Fecha de expiración de la cookie. Si esta seteada a la vez que Max-Age, prevalece Max-Age.

Domain: por defecto siempre están asociadas a la localización del documento actual, pero se puede definir un dominio para la misma.

Secure y HttpOnly no necesitan valores, solamente su presencia es valida para que el navegador las interprete. Ambas están asociadas con la seguridad.