El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

CICLO III:

Desarrollo de software

MisionTIC2022

El futuro digital es de todos | MinTIC

UNIVERSIDAD DEL NORTE
Vigilada Mineducación

Sesión 14:
Desarrollo Software

Método de autenticación basado en usuario y contraseña

Almacenamiento seguro de datos sensibles en un base de datos (contraseñas)

MisiónTIC2022

Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Aplicar los conceptos básicos sobre autenticación
2. Aplicar el método básico de autenticación basado en usuario y contraseña
3. Usar prepared statements para ejecutar queries.
4. Explicar las ventajas del uso de prepared statements
5. Aplicar el concepto de cookies y sesiones
6. Aplicar las funciones hash criptográficas
7. Usar las funciones criptográficas para almacenar contraseñas usando Salts

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

Método de autenticación basado en usuario y contraseña

Mision TIC 2022

Autenticación

Autenticación

Definición: La autentificación es la primera etapa del proceso de conexión de un usuario. Se puede definir como el acto que permite verificar que la persona que está ingresando al sistema es realmente quien dice ser. A la parte que se identifica se le llama probador. A la parte que verifica la identidad se la llama verificador.

Funcionamiento:

- Se aceptan las credenciales ingresadas por el usuario (usuario – contraseña).
- Se validan contra una base de datos, el sistema operativo, un servicio web, u otro mecanismo definido según el tipo de autenticación.

Autenticación

Métodos:
Se dividen en tres categorías:

- Sistemas basados en algo que el usuario sabe.
- Sistemas basados en algo que el usuario posee.
- Sistemas basados en una característica física del usuario o en un acto involuntario de este.

Características:

- Ser altamente confiable.
- Debe ser económicamente factible para la organización.
- Soportar con éxito algunos tipos de ataques.
- Ser aceptable para los usuarios, quienes serán los que lo utilicen
- Tener respuesta inmediata, directa, inteligente y sencilla ante cada situación.

El futuro digital es de todos MinTIC

UNIVERSIDAD DEL NORTE
Vigilada Mineducación

Autenticación - Tipos

Autenticación básica HTTP

Es el tipo más sencillo de autenticación web que existe. Solicita al usuario iniciar sesión con su nombre de usuario y contraseña. Sin embargo, la información se transmite usando codificación Base64. Esto quiere decir, que la información enviada no es cifrada y menos es segura. En teoría, la información podría ser interceptada por un tercero.

Autenticación de cliente HTTPS

HTTPS es muy utilizado en sitios web de e-commerce o en cualquier sitio en el que se tiene acceso a información confidencial. Proporciona un alto estándar de seguridad, ya que cada operación de intercambio de información entre el servidor y el navegador está cifrada y se envía a través de un canal seguro.

MisionTIC2022

Autenticación - Tipos

Autenticación basada en formularios

- Este tipo de autenticación se usa para recoger información de los visitantes que no requieren una alta necesidad de seguridad.
- Las encuestas, los formularios de contacto y las páginas de registro se hacen a menudo mediante la autenticación basada en formulario.
- El propósito fundamental de este tipo de autenticación es determinar qué campos del formulario se identifican como requeridos, para luego, asegurarse de que esos elementos se han llenado correctamente antes de pasar el formulario completo al destinatario.

Autenticación - Tipos

Autenticación mediante SMS

- Es el tipo de autenticación en dos pasos más utilizado.
- Este tipo de autenticación consiste en el envío de un código alfanumérico de único uso por medio de un mensaje de texto que el usuario introduce en la aplicación durante el inicio de sesión.
- A pesar de ser el tipo de autenticación más utilizado, no es el más seguro, ni el más fácil.
- Con respecto a seguridad, y teniendo en cuenta el auge del malware para dispositivos móviles, está entre los peores.
- En cuanto a usabilidad, puede llegar a resultar muy molesto para los usuarios el proceso de ir introduciendo el código al mismo tiempo que observan el mensaje.

Autenticación - Tipos

Autenticación mediante aplicaciones OATH TOTP

- Este tipo de autenticación utiliza aplicaciones de terceros para autenticar al usuario.
- La tecnología se conoce como OATH TOTP, y su funcionamiento es muy similar a la técnica de envío de códigos a través de SMS.
- En este caso el usuario utilizará una aplicación como Google Authenticator, LastPass Authenticator, o Latch para crear un código temporal de autenticación.

Autenticación - Tipos

Autenticación basada en factores biométricos

- Este tipo de autenticación se ha convertido uno de los favoritos por parte de los usuarios, debido a que es uno de los más seguros y cómodos en cuanto a usabilidad.
- Existen varios tipos de autenticaciones biométricas, entre las más utilizadas están el reconocimiento facial y dactilar, aunque también se encuentran soluciones que utilizan el iris o fisionomía.
- Su popularidad ha crecido en los últimos años. Esto se debe al fácil acceso del usuario a este tipo de tecnología y lo sencillo que es de usar. Por ejemplo, en el caso de reconocimiento dactilar en un móvil se reduce a que el usuario coloque su dedo en este.

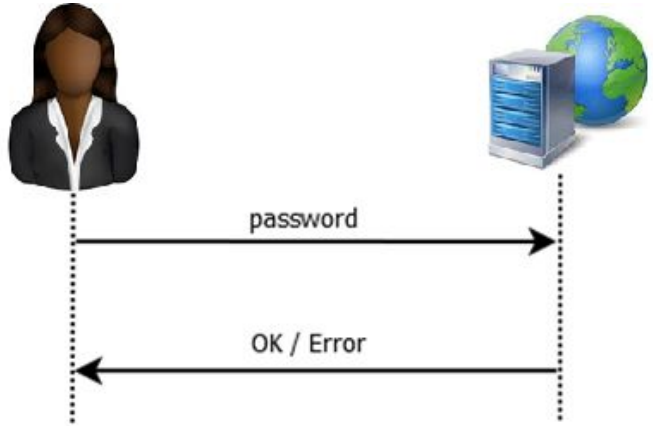
Autenticación basada en contraseña

Autenticación basada en contraseña

- El uso de contraseñas sigue siendo el método más utilizado para autenticación en la red. Esto se debe a que lo único que cada usuario debe recordar es su nombre de usuario y contraseña, en contraste con la molestia de tener que llevar consigo un certificado digital, token USB, tarjeta inteligente, disponer de hardware o software especializado, etc.
- Existen diferentes alternativas, cada una con distintas implicaciones en la seguridad del servicio y en la privacidad proporcionada a sus usuarios:
  - Transmisión simple de la contraseña: Autenticación básica.
  - Transmisión a través de canales cifrados: La contraseña se transmite por un canal cifrado.
  - Métodos desafío-respuesta basados en hashes: Protocolos de desafío-respuesta.
  - Métodos de conocimiento cero: El valor enviado al servidor para autenticarse no revela ninguna información sobre la contraseña.

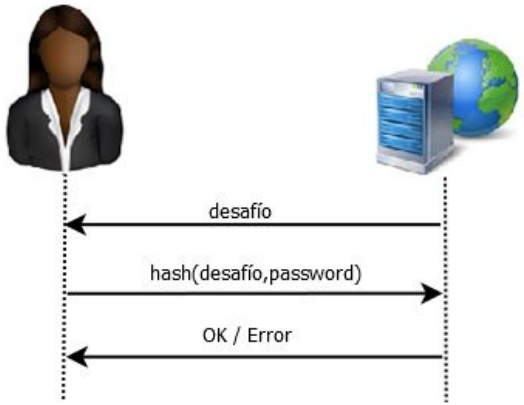
Autenticación basada en contraseña
Autenticación básica

- Se supone que tanto cliente como servidor conocen la contraseña (generalmente, el usuario la envía al servidor durante el registro inicial).
- En este método el cliente envía la contraseña y el servidor simplemente responde si la autenticación fue exitosa o no.



Autenticación basada en contraseña Desafío/Respuesta

- Una vez establecida la conexión, el servidor envía al cliente un desafío que cambia en cada autenticación, de tal forma que sólo un cliente legítimo (que sepa la contraseña) pueda contestar de forma correcta.
- Este método, podría decirse, es una mejora para la seguridad, ya que la contraseña no se transmite durante la autenticación. Se supone que tanto cliente como servidor conocen la contraseña (compartida durante el registro).
- Adicionalmente, la función hash refiere a una función para derivar llaves criptográficas basada en contraseñas, PBKDF2 por ejemplo.



Autenticación
Prepared Statements

- Prepared statements (sentencias preparadas) son plantillas utilizadas en consultas a sistemas de bases de datos en lenguaje SQL cuyos parámetros no tienen valores. Dichos valores, son reemplazados con variables o marcadores de posición, que no son sustituidos por los valores reales hasta estar dentro del sistema. Cuando las consultas son introducidas manualmente, los valores se asignan en el mismo momento de su ejecución.
- Las sentencias preparadas son utilizadas principalmente por seguridad cuando se trabaja con sistemas de gestión de bases de datos. El mayor problema de los métodos convencionales para acceder a las bases de datos en lenguaje SQL radica en la facilidad con la que pueden ser manipuladas.

Autenticación
Prepared Statements

- Estas plantillas no solo son utilizadas para la protección contra las inyecciones SQL: una vez analizada y compilada, un prepared statement puede ser reutilizado en el sistema de la base de datos tantas veces se desee, claro está, cambiando los respectivos datos.

- Por ende, cuando una tarea de repetirse en SQL, una y otra vez, las sentencias preparadas requieren muchos menos recursos y son más rápidas que las solicitudes manuales.

Autenticación
Prepared Statements - Fases

Fase 1: Preparación

Lo primero es crear una plantilla de sentencia. En lugar de los valores, a los parámetros principales se les asignan los marcadores de posición o parámetros de sustitución posicionales o variables bind.

```
INSERT INTO Producto (Nombre, Precio) VALUES (?, ?);
```

Fase 2: procesamiento de la plantilla en el DBMS

El sistema de gestión de bases de datos (DBMS) analiza sintácticamente la plantilla de sentencia, preparándose para su compilación como siguiente paso, y luego convirtiéndola en ejecutable.

Fase 3: ejecución

La plantilla procesada puede volver a utilizarse en el sistema de base de datos cada vez que se requiera o desee, con la única condición de que fuente de datos conectada proporcione correctamente los datos que reemplazarán a los marcadores de posición.

El futuro digital es de todos | MinTIC

UNIVERSIDAD DEL NORTE
Vigilada Mineducación

Autenticación - Cookies

MisiónTIC2022

Autenticación - Cookies

- Las Cookies son archivos que permiten guardar datos acerca de las preferencias del lado del cliente (usuario).
- Inicialmente almacenaban solamente información muy puntual. Por ejemplo, el lenguaje para ver una página web determinada (suponiendo que esté disponible en varios idiomas). Sin embargo, con el paso del tiempo, los datos almacenados en Cookies fueron creciendo para brindar una mejor experiencia de usuario.
- Para que la página pueda recordar la información de las preferencias, guarda las Cookies en la computadora (proceso gestionado por el navegador web; por ejemplo Chrome, Firefox, Edge, Opera, etc).
- Cuando se visita una página, ésta puede leer las Cookies registradas. Las Cookies guardarán cualquier dato que el servidor considere importante de recordar. Por ejemplo: la última fecha de visita una página, los productos cargados a un carrito de compras, los enlaces a visitados, etc.

Autenticación - Cookies

- Si una página solicita la creación de una Cookie, la Cookie se registra asociada a dicha página, y no puede ser consultada por otra página diferente.
- Las Cookies tienen un límite de tamaño, por lo tanto, cuando hay mucha información asociada a cada usuario, esta se guarda en el servidor, y en la Cookie se guarda solo un identificador (que le permita al servidor poder identificarla).
- Las Cookies no son malas. Su uso depende de los creadores de la página: la información que almacenan en Cookies y el uso que se le da a esta información.

Autenticación - Sesiones

Autenticación - Sesiones

- Una sesión es un mecanismo que permite que una aplicación asocie información con un cliente y que ésta se pueda recuperar con cada petición que hace el cliente.
- Normalmente esta información se almacena en archivos dentro de una carpeta del servidor (donde se guardan las variables de sesión y sus respectivos valores).
- Las sesiones sólo las puede crear y modificar el servidor. Toda la información de la sesión, se almacena en el servidor.
- Si no se finaliza, la sesión se mantiene hasta que el usuario cierra el navegador.
- Una sesión (al igual que una cookie) crea un archivo (donde se guardarán los datos).

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

Almacenamiento seguro de datos sensibles en un base de datos (contraseñas)

MisionTIC2022

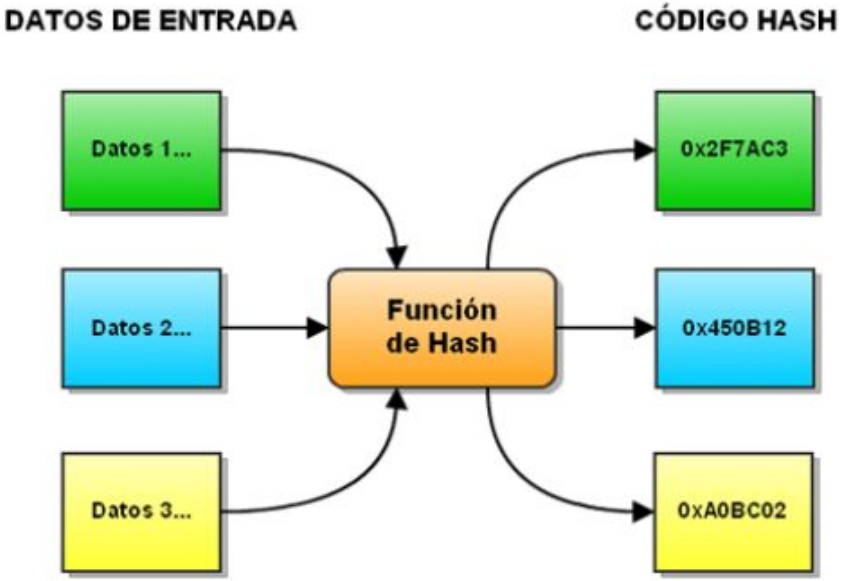
Criptografía

- Técnica utilizada para proteger documentos y datos.
- Funciona por medio del uso de cifras o códigos para escribir de forma confidencial o secreta en documentos y datos que transitan en redes locales o en internet.
- Rama inicial de las Matemáticas y que actualmente se utiliza en las ciencias de la computación, que por medio de métodos y técnicas logran hacer ilegible y por tanto proteger, un mensaje o archivo por medio de un algoritmo, usando una o más claves.

Funciones Hash

- Las funciones hash, conocidas también como “digest” o “de resumen“, son algoritmos que calculan y devuelven una cadena de texto alfanumérica de longitud fija la cual es calculada a partir de una entrada, que puede ser un texto o un archivo binario.
- Las hashs:
  - **Son funciones unidireccionales:** Teniendo el resultado y conociendo el algoritmo no debe ser posible revertir las operaciones para descifrar el dato de entrada.
  - **Son funciones determinísticas:** Siempre devuelven la misma salida para la misma entrada.
  - **Pueden generar colisiones:** Al ser resúmenes existe la posibilidad de que dos entradas totalmente distintas devuelvan el mismo resultado, es lo que se llama una “colisión“.

Funciones Hash



Funciones Hash Criptográficas

- Las funciones hash criptográficas son aquellas funciones hash que se utilizan en el área de la criptografía.
- Se caracterizan por cumplir propiedades que las hacen aptas y atractivas para su uso en sistemas que confían en la criptografía para proveerse de seguridad. Estas propiedades las hacen fuertes frente a posibles ataques maliciosos que quieran romper dicha seguridad.
- Una función hash criptográfica es un algoritmo matemático que convierte cualquier conjunto de datos en una nueva serie de caracteres con una longitud fija.
- Sin importar la longitud de los datos de entrada, el valor hash de salida siempre tendrá la misma longitud.

Funciones Hash Criptográficas

Propiedades:

1. El mismo mensaje siempre da como resultado el mismo valor hash.
2. El valor hash se calcula rápidamente.
3. No es posible tener dos mensajes con el mismo valor hash.
4. No es posible crear de forma intencional un mensaje que produzca un valor hash dado.
5. Por muy mínimo que sean los cambios en el mensaje, estos deberían cambiar el valor de hash resultante de manera significativa, de tal forma que parezca no correlacionado con el hash original.

Funciones Hash Criptográficas Usos

- Las funciones hash son utilizadas para almacenar contraseñas en una base de datos, evitando que se puedan descifrar inclusive si alguien lograra acceder a la base de datos.
- Otro uso fundamental de las hash es asegurar que un archivo no haya sido modificado durante el camino desde su envío: se hace el cálculo del hash antes de enviarlo, al recibirlo se realiza nuevamente el cálculo del hash, si el resultado no es el mismo implica que alguien ha capturado y modificado el archivo, es decir, se ha ejecutado un ataque de intermediario o man in the middle.
- Las funciones hash también se utilizan en los procesos de firma digital de documentos.

Funciones Hash Criptográficas

"Hashear" las contraseñas

- Hasta este momento, es claro que recuperar la contraseña de forma directa o mediante algoritmos reversibles no es buena idea. Por lo tanto, se estudiará la utilización de algoritmos irreversibles para reforzar la seguridad y es aquí donde entran a jugar un papel importante las funciones hash.
- Para verificar la contraseña de un usuario, se le aplica la función hash y se compara el resultado con el hash almacenado en la base de datos.
- Sin embargo, esta solución presenta algunos inconvenientes.
  - Una contraseña siempre genera el mismo hash, por lo tanto dos usuarios con la misma contraseña tendrán almacenado el mismo hash en la base de datos.
  - Son algoritmos computacionalmente muy rápidos, lo que conlleva a calcular el hash de un gran número de contraseñas por segundo con una GPU que muchas veces no tiene la capacidad.

Funciones Hash Criptográficas

Ataques

- Un **ataque de diccionario** utiliza un archivo de cadenas las cuales pueden ser utilizadas como contraseñas. Cada cadena en el archivo es “hasheada” y comparada con el hash de la contraseña objetivo. Si coinciden, la cadena es la contraseña buscada.

- Un **ataque de fuerza bruta** prueba cada combinación posible de caracteres hasta una longitud específica. Desde el punto de vista computacional, estos ataques son muy costosos y suelen ser los menos eficientes por tiempo de procesador, sin embargo, siempre hallan la contraseña.

- Un **ataque precomputado** consiste en crear una base de datos, que contiene un par de valores: la hash y su equivalente en texto plano. Esto una gran ventaja para el atacante, ya que existen servicios online que se dedican a alimentar constantemente su base de datos de hashes. De esta forma, un usuario puede hallar un hash buscando el valor en texto plano sin invertir tiempo en crackear la hash.

Funciones Hash Criptográficas

Hash con Salt

- Salt es un pequeño dato añadido que hace que los hash sean significantemente más difíciles de violar.
- Salt criptográfica es un dato que se utiliza durante el proceso de hash para eliminar la posibilidad de que el resultado pueda buscarse a partir de una lista de pares pre-calculados de hash y sus entradas originales, conocidas como tablas rainbow.
- Existe muchos servicios online que ofrecen enormes listas de códigos hash pre-calculados, junto con sus datos de entrada originales.
- El uso de salt hace muy difícil o imposible encontrar el hash resultante en cualquier listas.

Funciones Hash Criptográficas

Hash con salt

- Los hashs con salt, son como los hash de siempre con un añadido de seguridad: la salt que se le agrega.
- Salt es un número de dígitos aleatorios que se le agrega al hash ya sea al principio o al final. Con salt los hash dejan de ser comunes, lo que implica que es más difícil el decodificarlos, ya que se deberá probar tanto con cada hash, como también con cada salt y sus combinaciones.
- Utilizando salt se logra que las bases de datos de hash sean inútiles. Sin embargo, aunque se soluciona un inconveniente importante de las funciones de hash simples, el uso de salt no previene sobre ataques de fuerza bruta o de diccionario.

Funciones Hash Criptográficas con Python y HashLib

El módulo HahsLib.

- La librería estándar de Python contiene el módulo hahslib. Este módulo implementa una interfaz común a varios algoritmos seguros de hash.
- Se incluyen los algoritmos de hash seguros:
  - FIPS
  - SHA1, SHA224, SHA256, SHA384 y SHA512 (definidos en FIPS 180-2)
  - MD5 de RSA (definido en InternetRFC 1321).
- Los términos "hash seguro" y "resumen del mensaje" son intercambiables. El término moderno es hash seguro.

Funciones Hash Criptográficas con Python y HashLib

Algoritmos de hash:

Existe un método constructor para cada tipo de hash. Todos devuelven un objeto hash con la misma interfaz simple.

Ejemplo para cifrar con SHA-256 e imprimir el resultado en pantalla:

```
import hashlib
m = hashlib.sha256(b"mensaje")
print(m.digest())

b"]\xb3\xa6(\x06\x10!\x8a\x1c\x0b\xbf\xd9\x9f\x19\x82\x1e{\r$m'\x11\xc9\xf0\
xc7\x9c\x1f\x17\xfdF\xa09"
```

Funciones Hash Criptográficas con Python y HashLib

Al cambiar la m minúscula por la M mayúscula, el resultado sería totalmente diferente:

```
m = hashlib.sha256(b"Mensaje")
print(m.digest())

b'\xd2\xaf1q.\xad\x04\x14h\xc1\xc6\xfaLvV\\\\\xe0\xf9\xb1\x93\x10\x079\xffQ\
xb9\x07\xe4\xe3\x818'
```

Con solo cambiar un carácter, el hash cambia completamente.

Funciones Hash Criptográficas con Python y HashLib

La función `new()` retorna un nuevo objeto de la clase hash implementando la función (hash) especificada. Donde el primer parámetro es una cadena con el nombre de la función hash que se desea utilizar como “sha1”, “md5”, “sha224” y el segundo es la cadena a cifrar:

```
m = hashlib.new("sha256", b"mensaje")
print(m.digest())

b'\xd2\xaf1q.\xad\x04\x14h\xc1\xc6\xfaLvV\\\\\xe0\xf9\xb1\x93\x10\x079\xffQ\
xb9\x07\xe4\xe3\x818'
```

Funciones Hash Criptográficas con Python y HashLib

Algunos atributos constantes de los objetos hash retornados por los constructores:

- `Hash.digest_size`: El tamaño del hash resultante en bytes.
- `Hash.block_size`: El tamaño del bloque interno del algoritmo de hash en bytes.

Un objeto hash tiene los siguientes atributos:

- `hash.name`: El nombre canónico de este hash, siempre en minúsculas y siempre adecuado como un parámetro a `new()` para crear otro hash de este tipo.

Desde Python 3.4 no fue especificado formalmente, por lo que puede no existir en algunas plataformas.

Funciones Hash Criptográficas con Python y HashLib

Un objeto hash tiene los siguientes métodos:

- `hash.update(data)`: Actualiza el objeto de hash con el bytes-like object. Invocaciones repetidas son equivalentes a una única invocación con la concatenación de todos los argumentos: `m.update(a); m.update(b)` es equivalente a `m.update(a+b)`.

Distinto en la versión 3.1: El GIL de Python es liberado para permitir a otros hilos ejecutarse mientras ocurren actualizaciones de hash en datos con tamaños superiores a 2047 bytes cuando se usan algoritmos de hash suministrados por OpenSSL.

- `hash.digest()`: Retorna el resumen de los datos pasados al método `update()` hasta el momento. Este es un objeto de bytes de tamaño *digest_size* el cual puede contener bytes en el rango completo desde 0 a 255.

Funciones Hash Criptográficas con Python y HashLib

- `hash.hexdigest()`: Como `digest()` excepto que el resumen es retornado como un objeto de cadena del doble de largo, conteniendo sólo dígitos hexadecimales. Este puede ser usado para intercambiar el valor de forma segura en correos electrónicos u otros entornos no binarios.

- `hash.copy()`: Retorna una copia («clon») del objeto hash. Este puede ser usado para calcular eficientemente los resúmenes de datos compartiendo una subcadena inicial común.

Función de derivación de claves criptográficas (PBKDF2)

- PBKDF2 es un función de derivación de claves criptográficas, resistente a ataques de diccionario y tablas rainbow.
- Esta función se construye a partir de aplicar múltiples veces una función hash criptográficas, como por ejemplo SHA256 (por defecto).
- En Flask, podemos usar los siguiente métodos del paquete `from werkzeug.security`:
  - `generate_password_hash` para crear hashes resultado de aplicar PBKDF2.
  - `check_password_hash` para verificar una cadena generada por el método anterior.

Función de derivación de claves criptográficas (PBKDF2)

- Si se usa `generate_password_hash('2')`, se obtiene:
  `'pbkdf2:sha256:150000$Kc5ZhZvI$a749008a312f2b0b631b1253f0ba619d28982e874e35927f0315a1f151e72424'`

- Este cadena de salida se interpreta así:

- La cadena alfanumérica:
  `a749008a312f2b0b631b1253f0ba619d28982e874e35927f0315a1f151e72424` es el resultado.

- La cadena alfanumérica `Kc5ZhZvI` es el salt usado para generar el resultado.

- `pbkdf2:sha256:150000` significa que pbkdf2 aplica la función sha256 150000 veces para obtener el resultado.

Función de derivación de claves criptográficas (PBKDF2)

- Si se hace el llamado `generate_password_hash('2')` nuevamente, muy posiblemente obtendrá otra resultado distinto debido a que el salt se selecciona aleatoriamente.

- Tener en cuenta que se debe almacenar toda la cadena para realizar un chequeo posteriormente.

Función de derivación de claves criptográficas (PBKDF2)

- Para verificar `'2'` contra la cadena

  ```
  'pbkdf2:sha256:150000$Kc5ZhZvI$a749008a312f2b0b631b1253f0ba619d28982e874e359
  27f0315a1f151e72424'
  ```

- Se usa la método `check_password_hash` así:

  ```
  check_password_hash('pbkdf2:sha256:150000$Kc5ZhZvI$a749008a312f2b0b631b1253f
  0ba619d28982e874e35927f0315a1f151e72424','2')
  ```

- Esto debe dar como resultado `True`.

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

Ejercicios de práctica

Mision TIC 2022

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

¡GRACIAS

POR SER PARTE DE ESTA EXPERIENCIA DE APRENDIZAJE!

Mision TIC 2022