Técnico en

DESARROLLO DE SOFTWARE

Seguridad Informática

Instituto Von Neumann

Aplicaciones de la criptografía

Unidad VI

1. Protocolo SSL/TLS en comunicaciones HTTP (HTTPS)

Este es un tema clave para la seguridad web, es imperativo utilizar los certificados web para los sitios web. Más aun si los datos que se van a manejar son de características sensibles. En el caso de manejar tarjetas de crédito se debe utilizar PCI DSS (Payment Card Industry Data Security Standard).

HTTPS (HiperText Transport Protocol secured) se define como la versión de HTTP con una capa de seguridad basada en el TLS (Transport Layer Security), seguridad en la capa de transporte en español. Anteriormente se utilizaba SSL (Secure Socket Layer), pero se actualizo la versión y ahora es TLS, aunque usualmente se usan ambos términos como sinónimos.

Esta basado en la credibilidad que tiene un CA (Certificate Authority). El CA es un ente que da validez a la pagina web y de esta manera el browser el puede mostrar web como auténtico. Usualmente HTTPS utiliza el puerto 443 a diferencia de HTTP que va en el 80. Actualmente se debe utilizar la versión TLS 1.3.

La validación de una conexión segura se genera a través del procedimiento SSL/TLS Handshake. Aunque diferentes CA pueden utilizar versiones diferentes del procedimiento, existe una versión estándar en el cual se utiliza tanto el cifrado asimétrico como el simétrico, en este orden.

2. Autenticación

Es el proceso de verificar que un usuario sea quien dice ser. Existen 3 factores:

- **¿Qué tienes?:** tokens, correo, códigos QR
- **¿Qué sabes?**: contraseña, pin
- **¿Quién eres?:** huella digital, iris, identificación facial

La A2F es la autenticación utilizando 2 de los métodos mencionados.

Ejemplo:

- Usuario y contraseña (¿Qué sabes?)
- Token en aplicación de autenticación (¿Qué tienes?)

3. Cifrado de información sensible en base de datos

Cuando se almacenan datos de entidades en las bases de datos, a menudo se almacenan como texto plano o números. Usualmente los datos no tienen una capa de cifrado. Pero es importante que la información sensible sea almacenada en forma cifrada, de esta manera, aunque el sistema sufra un robo de información, los datos especialmente sensibles tendrán una capa extra de protección.

Se debe mencionar que, al igual que las demás prácticas de seguridad, se debe evaluar qué información amerita ser cifrada, dado que el proceso de encriptar y desencriptar consume recursos del sistema y aumenta el tiempo de espera de los usuarios. Por tal motivo, aplicarlo a todos los datos de la base de datos, es en general una mala práctica.

Personal Identifiable Information (PII)

Cualquier representación de la información que permita identificar a un individuo al que la se refiere directa o indirectamente se cataloga como PII. Como ejemplos se tiene la dirección de domicilio, numero telefónico, correo electrónico, nombre, Identificación personal (DPI o pasaporte). Pero también que permitan, mediante la combinación y análisis de datos, identificar a un individuo. Tal como un indicador

geográfico en un área junto con otra información como edad, raza, genero u otra descripción.

Niveles de cifrado de base de datos

Al cifrar una base de datos es posible implementarlo en varios niveles. Es importante mencionar que a mayor granularidad de cifrado el desempeño se ve más afectado.

- **Nivel de celdas**: a cada celda se le asigna una contraseña especifica.
- **Nivel de columnas o filas**: cada fila o columna posee su propia clave, es de las más comunes y que algunos proveedores ofrecen por defecto.
- **Nivel de tablas**: cifrado de las tablas.
- **Nivel de archivo**: en este nivel se encriptan los archivos individuales que contienen la información de múltiples tablas de una base de datos.

TDE (Transparent Data Encryption)

Es una arquitectura de seguridad que permite cifrar bases de datos, es utilizada por Oracle, MSSQL, MySQL, PostgreSQL y MariaDB además de otras. Se define como un método para la seguridad de datos en reposo y se implementa a nivel de archivos. Por lo cual protege ante robos de los archivos de la base de datos o robo de sistemas físicos de almacenamiento.

Es importante mencionar que este método no permite proteger la información ante ataques tipo inyección. Dado que cuando la información entra a nivel de aplicación, esta ya no está cifrada.

Al activar TDE también se encriptan los archivos de log y ya no se almacena la información en texto plano. Si la base de datos está en un sistema de replica automática, se debe activar TDE también en las replicas por separado.

La encriptación utilizada en TDE puede ser simétrica (AES o 3DES) o asimétrica (RSA). En el caso de la simétrica, la clave la debe almacenar el responsable de la base de datos y en el caso de la asimétrica, se almacena en el servidor como certificado.

Cuando se restaura una copia de seguridad de una base de datos con TDE, también se debe tener copia del certificado que permite desencriptar la información. Sin este certificado no es posible desencriptarla.

4. Cifrado de contraseñas en base de datos

Las contraseñas poseen una especial importancia cuando se considera la arquitectura a utilizar en la base de datos. Hay muchos vectores de ataque que puede dar acceso a la contraseña almacenada en la base de datos. En el caso de que la contraseña no se tenga cifrada, esto podría dar acceso al atacante y no solo al sistema en cuestión sino también a otros sistemas en donde el usuario utilice la misma contraseña o similar.

Por tales motivos es **imperativo que las contraseñas sean almacenadas en modo cifrado**, a nivel de celda. Utilizando algoritmos actualizados y seguros.

Las mejor practica para almacenar contraseñas tiene cuatro componentes:

1. **Función hash**: Es el algoritmo de cifrado de una vía que se aplica a las contraseñas. Es posible utilizar algoritmos como SHA, pero se recomienda utilizar algoritmos específicos como el Argon2id o PBKDF2.
2. **Factor de trabajo**: Es la cantidad de **veces** que se **aplica** la función hash sobre una contraseña. Se calcula como 2^factor.
3. **Salt**: Es un String que es **generado** de manera **aleatoria** y que se **agrega a la contraseña** antes de aplicar la función hash. El Salt debe ser **único para cada usuario.** La principal ventaja es que aun que dos usuarios utilicen la misma contraseña, esta resulta en diferentes Hash y por tanto aumenta la dificulta de adivinarla. Se debe **almacenar al lado** de la contraseña para que se pueda utilizar en el momento de autenticación.
4. **Pepper**: es una capa adicional que se agrega a las contraseñas, la diferencia con Salt es que esta adición es **compartida por todas las contraseñas** almacenadas. Usualmente se utiliza un algoritmo de cifrado **simétrico** sobre el hash generado y la llave es el Pepper. El Pepper debe ser almacenado

externamente a la base de datos. La rotación de las llaves es un factor adicional de seguridad.

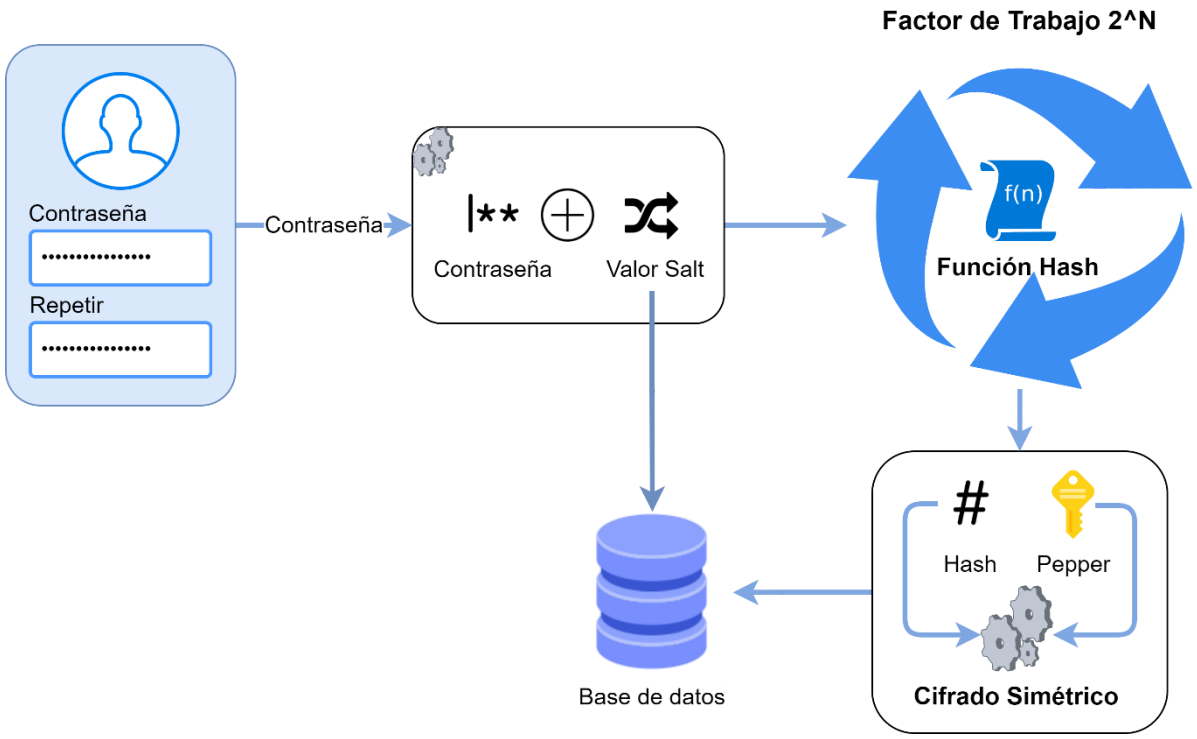

Imagen 1. Flujo para almacenar contraseñas.

Para autenticar a un usuario se debe seguir el **mismo flujo** que se utilizó al almacenar la contraseña, de esta manera obtener el mismo valor final y poder compáralo con el almacenado. Si estos valores coinciden se da por autenticado al usuario.

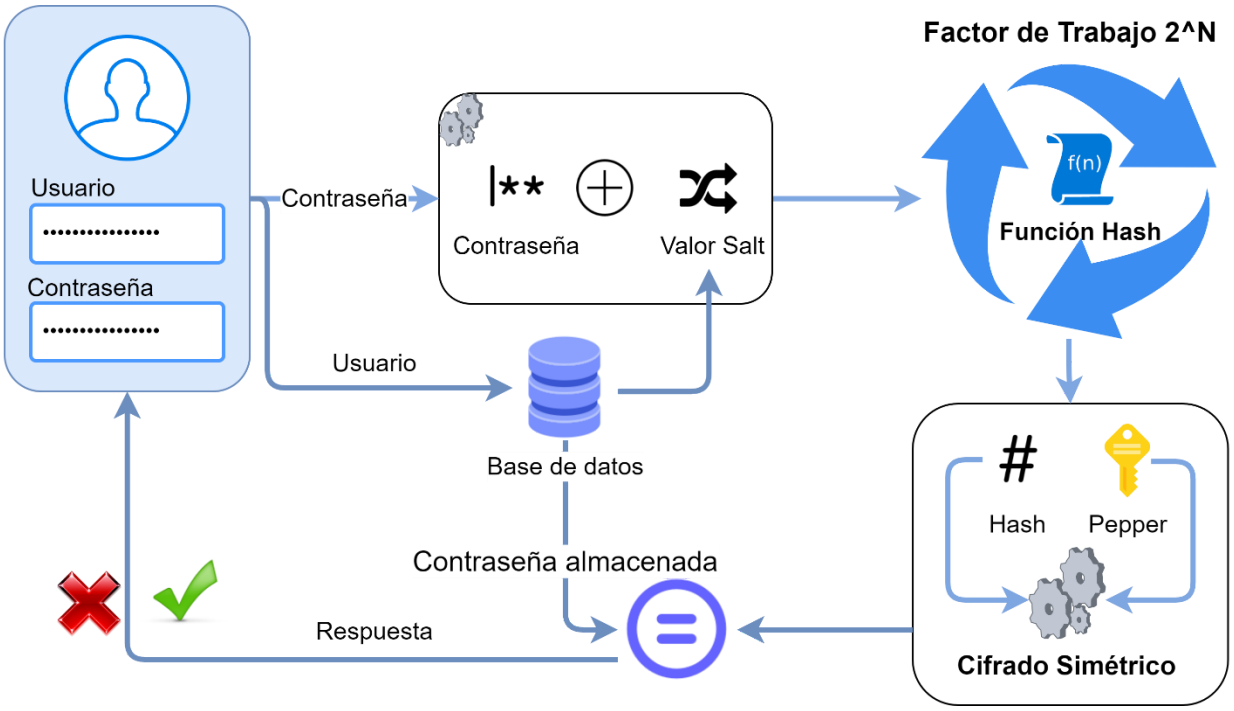

Imagen 2. Flujo para autenticar usuarios.

Hay 2 factores adicionales que puede incrementar la seguridad de las contraseñas utilizadas.

1. Verificar que la contraseña no este disponible en **diccionarios públicos** de contraseñas, los cuales son utilizados por atacantes.
2. Verificar que el **largo** de las contraseñas sea mayor a 7.

5. OWSAP

El OWASP Top 10 es un documento de concienciación estándar para desarrolladores y seguridad de aplicaciones web. Representa un amplio consenso sobre los **riesgos de seguridad más críticos** para las aplicaciones web.

- **A01:2021-Broken Access Control** El control de acceso se basa en aplicar una política tal que los usuarios no pueden actuar fuera de sus permisos previstos. Las fallas generalmente conducen a la divulgación, modificación o destrucción de información no autorizada de todos los datos o al desempeño de una función fuera de los límites de lo que el usuario esta autorizado.

- **A02:2021-Cryptographic Failures:** Este riesgo de seguridad surge cuando no se utilizan correctamente los algoritmos de criptografía o se utilizan de una manera que pueden ser vulnerados.
- **A03:2021-Injection:** Algunas de las inyecciones más comunes son SQL, NoSQL, comando OS, asignación relacional de objetos (ORM), LDAP e inyección de lenguaje de expresión (EL) o biblioteca de navegación de gráficos de objetos (OGNL). El concepto es idéntico entre todos los intérpretes. La revisión del código fuente es el mejor método para detectar si las aplicaciones son vulnerables a las inyecciones. Se recomienda la prueba automatizada de todos los parámetros, encabezados, URL, cookies, JSON, SOAP y entradas de datos XML.
- **A04:2021-Insecure Design:** Hay una diferencia entre el diseño y la implementación insegura. Diferenciamos entre fallas de diseño y defectos de implementación por una razón, tienen diferentes causas y soluciones. Un diseño seguro aún puede tener defectos de implementación que den lugar a vulnerabilidades que pueden ser explotadas. Un diseño inseguro no puede solucionarse con una implementación perfecta ya que, por definición, los controles de seguridad necesarios nunca se crearon para defenderse de ataques específicos.
- **A05:2021-Security Misconfiguration:** Entre las configuraciones inseguras debido a la mala implementación o negligencia en los parámetros se pueden encontrar las cuentas por defecto y las contraseñas genéricas como las más populares.
- **A06:2021-Vulnerable and Outdated Components:** Si el software es vulnerable, no es compatible o está desactualizado. Esto incluye el sistema operativo, el servidor web/de aplicaciones, el sistema de administración de bases de datos (DBMS), las aplicaciones, las API y todos los componentes, los entornos de tiempo de ejecución y las bibliotecas.
- **A07:2021-Identification and Authentication Failures:** La confirmación de la identidad de un usuario y el manejo de las sesiones es un punto critico para este

punto. Y el uso de métodos débiles para encriptar las contraseñas o su ausencia.

- **A08:2021-Software and Data Integrity Failures:** Las fallas en la integridad del software y los datos se relacionan con el código y la infraestructura que no protegen contra las violaciones de la integridad. Un ejemplo de esto es cuando una aplicación se basa en complementos, bibliotecas o módulos de fuentes, repositorios y redes de entrega de contenido (CDN) que no son de confianza.
- **A09:2021-Security Logging and Monitoring Failures:** esta categoría es para ayudar a detectar, escalar y responder a infracciones activas. Sin registro y supervisión, las infracciones no se pueden detectar. Registro, detección, monitoreo y respuesta activa insuficientes ocurren en cualquier momento.
- **A10:2021-Server-Side Request Forgery:** Las fallas de SSRF ocurren cada vez que una aplicación web obtiene un recurso remoto sin validar la URL proporcionada por el usuario. Permite que un atacante obligue a la aplicación a enviar una solicitud manipulada a un destino inesperado, incluso cuando está protegida por un firewall, VPN u otro tipo de lista de control de acceso a la red (ACL).

Referencias y contenido adicional para revisar:

1. http://es.ccm.net/contents/130-introduccion-al-cifrado-mediante-des
2. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
3. http://es.ccm.net/contents/criptografia-1747462277
4. http://es.ccm.net/contents/126-criptografia-de-clave-privada-o-clave-secreta
5. http://es.ccm.net/contents/127-sistemas-de-clave-publica
6. https://owasp.org/www-project-top-ten/
7. https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#peppering

Bibliografía

- Hodeghatta Rao, U., & Nayak, U. (2014). *The InfoSec Handbook.* New York: ApressOpen.

Descargo de responsabilidad

La información contenida en este documento descargable en formato PDF o PPT es un reflejo del material virtual presentado en la versión online del curso. Por lo tanto, su contenido, gráficos, links de consulta, acotaciones y comentarios son responsabilidad exclusiva de su(s) respectivo(s) autor(es) por lo que su contenido no compromete al área de e-Learning del Departamento GES o al programa académico al que pertenece.

El área de e-Learning no asume ninguna responsabilidad por la actualidad, exactitud, obligaciones de derechos de autor, integridad o calidad de los contenidos proporcionados y se aclara que la utilización de este descargable se encuentra limitada de manera expresa para los propósitos educacionales del curso.