Criptografía y Seguridad *(72.44)*

GUÍA 8: PROCESOS DE AUTENTICACIÓN - SOLUCIONES

Ejercicio 1:

a)Identifica los 5 componentes de un SISTEMA DE AUTENTICACIÓN basado en huellas digitales:

- *información de autenticación (A): la huella, o la fotografía de la huella*
- *información complementaria (C): el template*
- *funciones de complementación (F): el sistema que genera el template*
- *funciones de autenticación (L): dado template y huella, obtiene Verdadero o Falso*
- *funciones de selección (S): las de registro o enrolamiento.*

b) Da dos ejemplos de falsa aceptación que podrían darse en un sistema de autenticación por huellas digitales.
   a. *Clonación de template.*
   b. *Alteración de la comunicación entre escáner y base de datos*
   c. *Templates similares*

c)Da dos ejemplos de falso rechazo que podrían darse en un sistema de autenticación por huellas digitales.
   a. *Imagen dañada*
   b. *Scanner sucio*
   c. *Dedo lastimado→ esto se dijo en el caso del hijo de ANTONIO GRIMAU en 2010*[1]

«Cuando a un hospital llega un paciente con pérdida de conciencia, incapacidad de comunicarse o confuso, se hace el parte policial, entonces se le da intervención a la policía que está de guardia en el lugar o a la comisaría que corresponde al hospital», explicó a Diario Z el doctor Néstor Pérez Baliño, jefe de Gabinete del Ministerio de Salud porteño. Todo esto se cumplió en el caso de Rebolini Manso, aunque hubo demoras en los procedimientos siguientes.
Desde el hospital Fernández se le dio intervención a la Fiscalía de Instrucción 47, que ordenó que se obtuvieran las huellas dactilares del hombre y se enviaran a la Policía Federal y al Registro Nacional de las Personas (Renaper).
Ahora se sabe que el Registro Nacional de las Personas respondió que su archivo no está organizado en función de las huellas digitales, y que hacen falta un nombre o un número de documento. Con ese dato, el Registro localiza las huellas digitales y hace la comparación entre esa ficha y los registros tomados a la persona que permanece como un NN, un desconocido.
Ahora se sabe, también, que la calidad de las huellas digitales no era buena, que estaban desgastadas porque el hombre era músico y que algo parecido sucede con los trabajadores de la construcción o con personas que manipulan productos químicos, que «pulen» o «lavan» sus huellas. Otra infeliz coincidencia dejó a Rebolini Manso fuera del registro y de la posibilidad de encontrar a sus familiares mucho antes: el joven había tramitado sus documentos antes de 1991, año en que se incorporó un nuevo sistema de identificación denominado AFIS, Automatic Finger Identification System. De haber estado en esa base de datos, se hubiera identificado mucho antes.

d) ¿Podría ocurrir un robo de identidad en un sistema de autenticación biométrico? Justificar.
   "Aunque excede a la biométrica para abarcar información digital de cualquier tipo, la posible vulnerabilidad de las bases de datos que guardan la información personal de los individuos es un tema que puede generar desconfianzas.
   Con respecto a la posibilidad de robar información de una base de datos, el jefe del proyecto BioSec por Telefónica I+D, Orestes Sánchez Benavente, asegura que la vulnerabilidad "depende del diseño de las soluciones de autenticación. Es posible cifrar la información biométrica almacenada en una base de datos para evitar que pueda ser vista en caso de acceso fraudulento a la misma. Los canales de comunicación se pueden proteger también con técnicas criptográficas; y es posible combinar soluciones de autenticación remota y SmartCard, sin necesidad de almacenar la información biométrica en una base de datos. Estas alternativas van a ser estudiadas por el proyecto BioSec (www.biosec.org) que coordina Telefonica I+D"."

Ejercicio 2:

a) los caracteres deben ser dígitos. ("0" al "9")

[1] https://historico.diarioz.com.ar/#!/nota/nn-en-buenos-aires-el-caso-del-hijo-de-antonio-grimau-16725/

$P = \frac{T * R}{N}$ *donde T es tiempo, R es cantidad de passwords por segundo y N es el número total de passwords.*

$$P = \frac{31536000\ seg * \frac{16000\ pass}{seg}}{10^8} = 5045.76$$

El atacante puede explorar todo el espacio de passwords y más, es decir que Probabilidad es 100%

b) los caracteres deben ser alfanuméricos, pero sólo se usan letras mayúsculas (26 letras "A"…"Z" y "0" a "9")

$$P = \frac{31536000\ seg * \frac{16000\ pass}{seg}}{36^8} = 0{,}179 = 17{,}9\%$$

c) **Se agregan 10 bits de SALT a los casos anteriores.**

Si Alice y Bob ambos eligieran la misma password (por ej: `dontpwnme4`) el hash sería el mismo.
En el sistema se almacenarían:

username	hash
alice	`4420d1918bbcf7686defdf9560bb5087d20076de5f77b7cb4c3b40bf46ec428b`
…	…
bob	`4420d1918bbcf7686defdf9560bb5087d20076de5f77b7cb4c3b40bf46ec428b`

Si un atacante obtiene la password, la misma puede ser usada para acceder a todas las cuentas que tienen el mismo hash. Además, ya sabe que sólo están formadas por letras y números, por lo que el ataque es más rápido, tanto por **fuerza bruta** (utilizando caracteres aleatorios explorar todas las combinaciones) como por **ataque de diccionario** (usando listas de palabras)
Por otra parte, el atacante podría disponer de **hash tables y rainbow tables.**

Hash tables: bases de datos de hashes previamente calculados sobre palabras de un diccionario o sobre cadenas aleatorias. Son de búsqueda rápida una vez hechas, ya que permite que, dado un hash, se busque en la tabla directamente cuál puede ser la password asociada. Para ahorrar tiempo y espacio el atacante puede elegir procesar primero las passwords más comunes.
Rainbow tables: ocupan menos espacio pero requieren repetir el algoritmo de hash varias veces, entonces es más lento, aunque suelen ser más efectivos.

Para mitigar el daño de los ataques anteriores, se puede agregar un salt, que es un valor generado por alguna función criptográficamente segura, que se agrega como entrada a la función de hash, para crear hash únicos para cualquier entrada. Si dos usuarios usan la misma password, con el agregado del salt, ya no se guardará el mismo hash.
Para el caso anterior, de Alice y Bob usando la misma password `dontpwnme4:`

Password de Alice: `dontpwnme4`
Salt `f1nd1ngn3m0`
Input para el hash: `dontpwnme4f1nd1ngn3m0`
Hash (SHA-256): `23909a604f3b5cbd8a3b802f3345d58ad852cba420b2e481f1d36b8f0e557efb`

Password de Bob: `dontpwnme4`
Salt `f1nd1ngd0r`
Input para el hash: `dontpwnme4f1nd1ngd0r`
Hash (SHA-256): `7ae78cf28a08ccbb364d93977e7ffe5b4256f36dc7d5b4c3bf31d597f8bd72ab`

Distintos usuarios, misma password, distintas salts, distintos hashes.

Por lo tanto, volviendo a nuestro problema, si agregamos 10 bits de salt, ahora las combinaciones que debería probar el adversario serían
$(26+10)^8.2^{10}$= 208827064576.1024 = 2.1383891e+14
Y en lugar de adivinar 16000 pass por segundo, sólo podrá hacer 16000/1024

$$P = \frac{31536000\ seg * \frac{15265\ pass}{seg}}{36^8} = 0{,}00017 = 0{,}017\%$$

En la práctica, se almacenan el nombre de usuario, el salt y el hash. Cuando el usuario ingresa escribiendo su password, el sistema le agrega el salt correspondiente a ese usuario y calcula el hash. Si el hash coincide con el almacenado, el acceso está permitido.
Si el atacante tiene acceso a los salts, deberá probar en cambio $(26+10)^8$.número de salts.
Para mejorar el uso del salt se pueden tener en cuenta las siguientes opciones:

- Almacenar salt y hash en forma separada.
- Que el salt tenga la misma longitud del hash.
- Que el salt se calcule a partir del nombre de usuario (controlando que sigan siendo salts únicos). Eso permitiría no tener el salt guardado, sino que se calcula cada vez.
- Encriptar el hash con alguna clave secreta.