Estudio de Soluciones de Gestión Personal de Contraseñas

La cantidad de contraseñas que debe memorizar un usuario ha ido creciendo y creciendo sin límite durante los últimos años. Obviamente había que dar respuesta a esta situación y se encontraron (herramientas para la gestión de Identidades y accesos, *single sign-on*, etc.) pero aún así, hoy y siempre, se presentan casuísticas frente a las cuales, las más serias y, por ende muchas veces, complejas soluciones de la industria no son capaces de adaptarse, o lo hacen de forma demasiado prolongada en el tiempo, o incluso -por cuestiones del tamaño, recursos y coste para la empresa implicada-, no sea viable su despliegue para solventar el problema. En esos casos hay que optar por soluciones ágiles y de fácil despliegue. Y si, además, el coste es reducido o incluso cero, aún como solución temporal, pueden ser una gran alternativa. Este artículo muestra los resultados obtenidos tras un análisis realizado sobre un conjunto de herramientas de Gestión Personal de Contraseñas basadas en Open Source, es decir, con el código disponible y un coste cero de licencias.


Daniel Fernández Bleda / Ángel Puigventós Gràcia

Cada vez más, la cantidad de contraseñas que debe memorizar un usuario ha ido creciendo y creciendo sin límite durante los últimos años. En algunas estadísticas se han encontrado casos de personas que emplean hasta 40 pares de identificadores y contraseñas en su trabajo diario.

Obviamente, había que encontrar soluciones a esta situación y se encontraron: sistemas de Single Sign-On, la más nombrada últimamente Gestión de Identidades, etc.

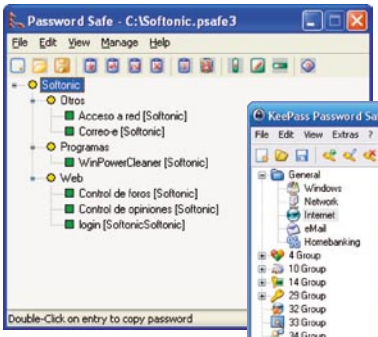

Figura 1. Password Safe

Pero aún así, hoy y siempre, se presentan casuísticas a las cuales, las más serias y, por ende muchas veces, complejas soluciones no son capaces de adaptarse o, incluso, se adaptan de forma demasiado prolongada en el tiempo. Algunos casos pueden ser, por ejemplo, administradores de redes que deben recordar multitud de contraseñas difíciles de gestionar centralizadamente (y en las que resulta un riesgo muy grande emplear contraseñas únicas y compartidas), el tiempo durante el cual se llevan a cabo procesos de integración y despliegue de soluciones de centralización, unificación o gestión de identidades, y no hay que olvidar empresas en las que, por su tamaño o recursos, no sea viable el despliegue de una solución integrada para la solución de este problema cuyo coste pueda no ser abordable.

En esos casos hay que optar por soluciones ágiles y fácil despliegue y si, además, el coste es reducido o incluso cero, aun como solución temporal, puede ser una gran solución.

Este artículo muestra los resultados obtenidos tras un análisis realizado sobre un conjunto de herramientas de Gestión Personal de Contraseñas basadas en Open Source, es decir, con el código disponible y un coste cero de licencias.

Estos programas permiten mantener, de forma segura (con cifrado fuerte), un repositorio personal de contraseñas, de forma organizada y particular. Estos programas basan la protección de dicho repositorio en el hecho de emplear una "*passphrase*" lo suficientemente robusta como para que no pueda ser obtenida, pero siendo únicamente ésta la que se ha de recordar, pudiéndose así realmente olvidar todos los identificadores y contraseñas que se almacenan en su interior. De esta forma estas contraseñas pueden ser realmente complejas y difícilmente obtenibles mediante ataques de *password cracking*.

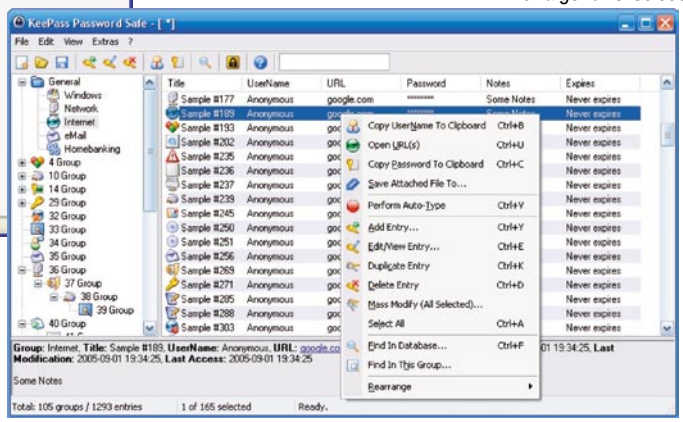

Figura 2. Keepass: pantalla principal

Alguno de estos programas puede ejecutarse incluso en dispositivos PDA, confiriéndoles una flexibilidad mucho mayor ofreciendo una disponibilidad total de las credenciales cuando nos desplazamos fuera del puesto de trabajo.

Algunas de las características claves analizadas en cada uno de los programas han sido estas:

- Algoritmos de cifrado utilizados para almacenar la información en disco o transferirla en red.
- Tipo y formato de la base de datos.
- Posibilidad de ocultar visualmente la contraseña.
- Posibilidad de generar llaves aleatorias de calidad[1].
- Tiempo de espera: cierra la aplicación automáticamente para evitar descuidos.
- *Backup*: posibilidad de guardar ficheros *backup*.
- Posibilidad de importar/exportar ficheros y compatibilidad con otros programas.
- Sistema de actualizaciones: automático, por aviso o no dispone.
- Facilidad de uso.
- Sistema de eventos: posibilidad de ver un historial de accesos a las contraseñas.

Keep It Secret! Keep It Safe! (KisKis)

Creada en Java, ofrece soporte para cualquier plataforma que tenga soporte para este lenguaje. Su funcionamiento es bastante sencillo, pero no deja de tener algunos errores de programación a nivel de la gestión de guardado de ficheros, ya que no es capaz de crear un nuevo fichero, solo modificar uno existente, por lo que hay que crear uno vacío y usarlo.

Se usa una única contraseña para acceder al fichero cifrado, la cual es la que se encarga de cifrarlo o descifrarlo, por lo que el resto de los datos quedan a merced del nivel de esa contraseña. Ella es la que se encarga de cifrar el contenido de las entradas con el algoritmo seleccionado; de entre los ofrecidos, Rijndael y Blowfish son los más recomendables.

Entre sus características a la hora de poner una llave en el sistema, éste nos dará el nivel de calidad según los tipos de caracteres introducidos, letras, dígitos, caracteres especiales, etc.

JPasswords

Otra herramienta creada en Java aparentemente más sencilla, pero igual de potente. A diferencia de "KisKis", aquí se debe introducir una contraseña de nivel suficientemente buena; si no, éste no nos dejará continuar. Estas herramientas evalúan el nivel de la contraseña en función del tipo de caracteres introducidos así como su orden.

Su entorno de interfase es poco agradable y recuerda al puro estilo visual de Java/Sun. Por lo demás, está pensado para ser sencillo y funcional sin tener que recurrir a menús y pestañas engorrosas.

La manera de ejecutar el JAR bajo Windows implica estar siempre pendiente de la máquina virtual java para poder iniciar el programa.

[1] Para calcular los valores de entropía y secuencialidad, se ha partido de generar 3 cadenas aleatorias en cada programa, con una longitud de 2 ^ 8 caracteres. Es decir, que una cadena de 256 caracteres únicamente con el valor de "A", el resultado seria de 0 bits por byte. Mientras que una cadena completamente aleatoria rozaría los 8 bits por byte.

Entropía: es la densidad de información contenida en cadenas o ficheros. Si un archivo es muy denso en información de valores aleatorios, no podrá ser comprimido. Por lo que los ficheros de compresión tienen valores altos de entropía. Por lo contrario, códigos fuente en C o textos tienen un valor bajo, cerca del 4,9.

Nombre	Keep It Secret! Keep It Safe! v0.19.2		JPasswords v0.3.1		Password Safe v3.06		Keepass v1.06	
URL	http://kiskis.sourceforge.net/		http://jpws.sourceforge.net/jpasswords.html		http://passwordsafe.sourceforge.net/		http://keepass.sourceforge.net/	
Lenguaje	JAVA		JAVA		C++		C++	
Open Source	Sí, mediante CVS		Sí		Sí		Sí	
Algoritmos	Rijndael, Blowfish, CAST		Blowfish		SHA-256 y Twofish		Rijndael o Twofish	
Base de datos	Fichero Único		SHA-1, Blowfish		Fichero		Fichero único	
Ocultación de la Contraseña	Seleccionable		Seleccionable		Seleccionable		Seleccionable	
Generación de Llaves	Parametrizable	Calidad	Parametrizable	Calidad	Parametrizable	Calidad	Parametrizable	Calidad
	-	-	-	-	Si	5'8 bits por byte (Normal)	Sí	5'4 bits por byte (Normal)
Clipboard	No		Usuario y Contraseña (Auto-borrado en 30 segundos)		Usuario, Contraseña, Notas. (Borrado bajo petición)		Usuario y Contraseña. (Auto-borrado en 10 segundos)	
Expiración de la Contraseña	365 días		Nunca, por defecto (1mes a 5 años)		Nunca, por defecto		Nunca, por defecto	
Tiempo de Espera de la Sesión	5 minutos por defecto		5 minutos, por defecto		5 minutos, por defecto		Nunca, por defecto (configurable, en segundos)	
BackUp	Automáticos o bajo petición		Bajo petición		Sí		Automático	
Posibilidad de exportar ficheros	Importar	Exportar	Importar	Importar	Importar	Exportar	Importar	Exportar
	Si	CVS, HTML, XML, User-Defined	CVS	CVS, Fichero Cifrado	Texto Plano, XML, Keepass	Versión 1 y 2, Texto plano, XML	CVS, Importación*	TXT, HTML, XML, CVS
Actualizable	Configurable, automático al iniciar		No, pero permite mirar las noticias nuevas		No		Verificación bajo consulta	
Facilidad de Uso	Sencillo		Bastante sencillo		Buena		Normal	
Sistema de Eventos (Logs)	Historial de accesos a las secciones.		Historial de accesos a las secciones		Historial de accesos y cambios		Sí	
Curiosidades	-		WIPE (Borrado seguro de ficheros)		Existe una versión para WinCE ideal para PDAs		Permite el uso de "Plugins". Entre ellos, idioma Catalán y Castellano	

* Importa ficheros de otros programas, entre ellos: Code Wallet, Password Safe y Personal Vault. Y Keepass Database.

Password Safe

A diferencia de las otras dos herramientas, ésta está escrita en C++. Pero también es sencilla, por lo que no se necesita pasar más de 15 minutos para saber cómo funciona toda ella.

Se aprecia el uso del algoritmo de cifrado Twofish, la variante más nueva y segura de Blowfish, finalista en el NIST para ser el posible AES (finalmente quedé tercero tras Rijndael y Serpent).

Se puede apreciar en el historial del desarrollador como la herramienta ha sufrido una evolución hacia mejoras constantes, por lo que cabe esperar nuevas opciones en el futuro y soporte para la corrección de posibles errores.

Creada inicialmente por Bruce Schneier (autor de Blowfish y Twofish con otros) en Counterpane.com.

Keepass

La herramienta más popular para la gestión de contraseñas debido, quizás, a su buena interfaz y estructura de las contraseñas. Cuenta con varios clones: Java clone of Keepass, como bien indica su nombre, hecho en Java; Keepass for Smart Devices, para PDAs WinCE 2003 y 5; Keepass Micro Edition, para PalmOS; y KeepassX versión para Linux y MacOS X.

Ofrece una imagen cuidada, su gestión es buena y sencilla, por lo que agrada trabajar con él, ver los datos, etc.

Asímismo, el poder generar *plugins* hace que se pueda modular el programa a nuestras características específicas; eso sí, al ser bastante completo deja poco a la imaginación.

Uno de los *plugins* recomendables es el de la integración de Keepass en la barra del IExplorer.

VALORACIONES

Desde el punto de vista técnico en función de los algoritmos de cifrado, la mayoría cumplen los requisitos de formar parte del AES, exceptuando Sleutel que utiliza una versión de 3DES (TripeDES) o Blowfish que no fue presentado en esa convocatoria pero que actualmente goza de uno de los mejores puestos en los algoritmos de cifrado simétrico.

El tener que utilizar una máquina virtual de Java para poder ejecutar algunos de los programas, supone descargarse este paquete y en el mejor de los casos, ejecutar el programa de forma transparente cosa que no sucede con JPasswords, que requiere ser forzado desde la VM JAVA para su ejecución. Otro punto negativo es el hecho de que se pierde algo de rendimiento al procesar el código Java; en nuestro caso no se precisa importancia a esta pérdida.

Por su parte, Keepass cuenta con muchas opciones configurables según nuestras preferencias así como soporte de lenguajes y *plugins* para facilitar el acceso al mismo, aunque esto podría suponer una vía de acceso sin autorización. Cuenta también con muchos proyectos no oficiales de su versión sobre otros sistemas operativos, así como para PocketPC (ideal para llevar siempre con nosotros las llaves). No es el único que tiene este soporte para PDAs; Password Safe dispone de una versión oficial, pero no goza de mucha continuidad de desarrollo.

En función a su forma de organizar, trabajar y acabado visual, tanto Password Safe como Keepass ofrecen los mejores resultados, aunque este apartado podría ser el más discutido, ya que afecta a valoraciones subjetivas por los gustos de cada uno.

Por todo lo valorado el que mejor nos ha parecido y se adapta a las necesidades es Keepass.

RECOMENDACIONES DE USO

Como resumen, se plantean las siguientes recomendaciones sobre los programas analizados:

• Puede ser recomendable mantener una copia de seguridad del fichero en un soporte extraíble (como por ejemplo un *token* USB).

• Los programas basados en Java requieren del uso de las Máquinas Virtuales para que puedan funcionar correctamente. Aunque parezca que ocupan bastante poco, hay que contar con el uso de estas VM. La ventaja es que pueden funcionar en todos los sistemas con soporte Java.

• Cualquier llave de más de 8 caracteres, conteniendo valores alfanuméricos mas signos de puntuación y similares, nos proporciona un nivel de seguridad muy bueno contra ataques de fuerza bruta y/o diccionario. El concepto más importante es la no necesidad de recordarlas mentalmente, podemos emplear longitudes mayores, de 10 a 16 caracteres (cuando sea posible).

• Ante el hecho de que toda la seguridad se base, normalmente, en una contraseña maestra es crítico que esta contraseña tenga una longitud elevada (mayor de 16 caracteres) pero ha de ser fácil de recordar (por ejemplo: "3l p3rr0 de R0qu3 n0t13n3 r4b0"). Teniendo presente que el fichero de claves puede ser robado, es imprescindible que su robustez sea muy elevada.

• Navegadores como el IExplorer permiten por defecto el uso de exportar el contenido del portapapeles mediante "javascripts" embebidos en las páginas HTML. Por lo que se recomienda, si se copia la contraseña en el portapapeles, que ésta sea destruida tras su uso así como no visitar ningún sitio mientras se dispone a utilizarla. ■

Daniel Fernández Bleda
Consultor en Seguridad/Socio
CISA, CISSP, CHFI, OPST/OPSA, ISO27001 Lead Auditor
dfernandez@isecauditors.com

Ángel Puigventós Gràcia
Analista en Seguridad, CEH
apuigventos@isecauditors.com
INTERNET SECURITY AUDITORS