La Ardua Tarea de Asegurar la Identidad

Por Gunnar Wolf

Gunnar Wolf es administrador de sistemas para el Instituto de Investigaciones Económicas de la UNAM y desarrollador del proyecto Debian GNU/Linux.
http://gwolf.org

Una de las tareas que realizo dentro del Proyecto Debian es la de actuar como curador de los llaveros de identidad criptográfica que identifican a los participantes del proyecto, y con que realizamos prácticamente cualquier acción (desde mandar un mensaje a las listas moderadas hasta subir nuevos paquetes) con verificación automática de identidad.

A últimas fechas tuve la oportunidad de participar en el Congreso de Seguridad en Cómputo que organiza UNAM desde hace más de quince años [1], presentando algunas de las experiencias y aprendizajes derivados de esta tarea [2]. Dicho trabajo me lleva a algunas reflexiones que considero que pueden ser de interés para los lectores de SG.

¿Por qué no contraseñas?

El uso de pares usuario-contraseña para validar la identidad de un usuario y determinar si tiene permiso de utilizar determinado recurso se ha utilizado en el cómputo por lo menos desde la década de los sesenta, y en el desarrollo cultural humano, hay documentos que confirman su uso por lo menos desde la época del Imperio Romano. Nuestros usuarios las conocen y comprenden su funcionamiento, y nadie dirá que vamos contra las prácticas más comunes al emplearlas.

Sin embargo, tenemos que estar conscientes de la relativa fuerza (o debilidad) de este esquema: ya desde 1979 hay estudios que demuestran que son un mecanismo débil y muy falible de autenticación [3]. ¿Vamos a exigir a nuestros usuarios "contraseñas fuertes"? Esto es, ¿tenemos una política que requiera que toda contraseña sea (por ejemplo) una cadena de 10 o más caracteres, que no formen palabras, incluyendo mayúsculas, minúsculas, números y caracteres especiales? Malas noticias: si la tenemos, estamos básicamente asegurando que la contraseña será escrita en un papelito que seguramente vivirá pegado al monitor o debajo del teclado — Mucho menos seguro que una contraseña "sencillita" y memorizable.

Pero, por otro lado, ¿qué tanta expectativa de seguridad tenemos si permitimos contraseñas memorizables? La imagen que acompaña este artículo muestra las contraseñas más frecuentemente utilizadas, con tamaños asignados según su frecuencia relativa, tomadas de una muestra de un millón de cuentas (Mark Burnett, 2011). Uno de los puntos más preocupantes: Las 10,000 contraseñas más comunes corresponden al 98.8% del millón de cuentas. ¿Qué tan impredecibles esperamos que sean nuestros usuarios, o nosotros mismos?

En el caso particular de Debian, los recursos a que tendrán acceso los desarrolladores una vez autenticados son enormes. Por poner un simple ejemplo, al subir una nueva versión de un paquete, éste puede instalarse en millones de computadoras, ejecutando con privilegios de administración. Las contraseñas simplemente no son suficiente protección.

Más allá de las contraseñas

Podemos encontrar numerosos esquemas de autenticación que nos llevan más allá de las contraseñas. Casi todos los esquemas siguen los lineamientos de la autenticación multifactorial: si la contraseña es "algo que yo sé", debo combinarla con "algo que yo tengo" y/o con "algo que yo soy" para obtener una autenticación fuerte.

Un ejemplo con el que seguro han interactuado son los tokens (pequeños dispositivos que presentan un número de seis u ocho dígitos que cambia típicamente cada minuto). Estos dispositivos son claramente "algo que tengo" físicamente, con todo lo bueno y malo que esto conlleva: siendo un objeto físico, no puedo copiarlos. Siendo un objeto que cambia dinámicamente, no puedo memorizarlo ni predecirlo. Si dejé el token en casa o lo perdí junto con mis llaves, no podré identificarme ante el recurso en cuestión.

Por otro lado, se ha popularizado el confiar en los datos biométricos: presentar a la computadora una prueba inherente a mi cuerpo, "algo que yo soy". Ejemplos de esto son lectores de huellas digitales, sistemas de reconocimiento de voz, escáneres de retina —e incluso el reconocimiento fotográfico que podría hacerse con una webcam cualquiera. Los datos biométricos sobreviven a largo plazo (no necesariamente de por vida, pero sí a largo plazo; mi envejecimiento o un cambio radical de estilo podrían causar que una foto de mi webcam no me reconozca), no puedo olvidar traerlos conmigo, no puedo copiarlos, pero por otro lado resulta imposible pedirle a otra persona que haga

algo a mi nombre. Y claro, de los ejemplos presentados, el reconocimiento por webcam sería el más débil. Una persona muy parecida a mí podría pasar bajo el umbral de decisión y reemplazarme.

Autenticación por criptografía de llave pública

En el caso de Debian (y de la gran mayoría de las comunicaciones cifradas del mundo) hemos optado por una versión ligeramente debilitada de emplear "algo que tengo": para interactuar con el proyecto, mi interacción debe ir firmada criptográficamente por una llave que forme parte de uno de los llaveros del proyecto —que uno de los curadores haya aceptado.

Un par de llaves criptográficas es, a fin de cuentas, un par de archivos (uno para la llave pública, otro para la llave privada) en mi computadora, y de ahí que sea únicamente una versión débil de "algo que tengo": un archivo puede ser copiado, lo que se traduce en la imperiosa necesidad de ser muy diligente protegiendo el medio en el que éste se almacena; si este par de archivos cayera en manos de terceros, quien lo tuviera podría firmar cualquier archivo de forma indistinguible de lo que haría yo. Claro, un certificado siempre debe estar protegido por una contraseña como primerísima línea de defensa —pero la seguridad de la llave privada sencillamente no puede tomarse a la ligera.

Además de Debian, el uso de certificados para autenticación es muy común en los principales proyectos de software libre. Pero su uso no permanece únicamente en este ámbito: La Firma Electrónica empleada por el Servicio de Administración Tributaria (Hacienda) en México, e incluso por el Sistema Integral de Administración Escolar de la UNAM, ambos ejemplos de sistemas donde el riesgo de mal uso por una insuficiente identificación es demasiado grande.

Aquí puede comenzar a apreciarse por qué se elige este factor "algo que tengo" debilitado. Por la distribución de material. Si Debian, el SAT o la UNAM operarán con tokens hardware, tendrían que repartir un dispositivo a cada participante, lo cual conlleva un costo nada trivial. Tendrían que establecerse mecanismos de restablecimiento ante tokens perdidos o dañados. Y si la tecnología avanza y hace obsoleta a una generación de tokens, hay que reemplazar a todos aquellos que sigan en uso. Manejando un par de llaves, este gasto se mitiga fuertemente.

¿Y qué es esto del par de llaves? Muy en resumen, la criptografía de llave pública se basa en funciones matemáticas unidireccionales (esto es, relativamente fáciles de calcular en un sentido, pero muy difíciles de calcular en sentido contrario) para las que existen parejas de valores. Por ejemplo, una operación que podemos realizar a ojo es multiplicar dos números primos de dos dígitos:

$$23 \times 47 = 1081$$

Sin embargo, si queremos factorizar 1081, la tarea resulta mucho más difícil —se vuelve un ataque de fuerza bruta. Esto es, en resumen y de forma un tanto simplificado, el funcionamiento de la familia de algoritmos más ampliamente utilizada, RSA — Aunque, claro, no con números de dos dígitos, sino de cerca de mil (entre 2^{2048} y 2^{4096}). De estos números se puede derivar una llave pública, que puede ser presentada públicamente para que cualquiera pueda comunicarse de forma segura con su propietario, y una llave privada, que debe ser bien resguardada y jamás debe caer en manos de terceros (y en caso de hacerlo, debe revocarse inmediatamente).

Certificados e identidad

Tener este par de números no asegura, sin embargo, la identidad. ¿Cómo he de saber que la llave pública que estoy empleando corresponde realmente a la contraparte con que deseo comunicarme y no a un impostor? Agreguemos restricciones: yo no tengo trato directo con mi contraparte.

Aquí es donde entran en juego los certificados. Un certificado es un archivo que contiene la información necesaria de una llave pública junto con la firma de uno o varios (dependiendo del modelo empleado) proveedores de confianza, aseverando que verificaron la identidad del poseedor de esta llave, y asegurando que corresponde con quien dice ser.

El modelo de confianza más difundido en Internet es PKI (Public Key Infrastructure), en que una serie de empresas dedicadas a la creación de certificados goza de nuestra confianza universal. Este modelo es de fácil adopción, y no en balde sobre él se construyeron los modelos de cifrado SSL y TLS; prácticamente todo el tráfico cifrado comercial de Internet los maneja, pero su naturaleza centralizada lo ha llevado a una serie de graves fallos de confianza [4].

El proyecto Debian, entre otros muchos, emplea un modelo descentralizado, par-a-par, llamado llavero de confianza (Web of Trust). En este caso, en vez de depender de una autoridad central, cada uno de los participantes del llavero puede certificar y ser certificado por quien conozca. De este modo, sin que dos personas se conozcan, pueden establecer qué tan fuerte es el camino de confianza que los une dentro de un universo determinado.

El espacio me impide entrar más a detalle sobre este tema. Les refiero a un análisis estadístico, automático y actualizado periódicamente, del comportamiento del principal directorio de llaves PGP [5], que al día de hoy cuenta con más de 4 millones, y me comprometo a abordar algunos aspectos interesantes de la gestión de estos llaveros en números siguientes de SG.

Referencias

[1] Congreso en Seguridad de Cómputo. UNAM. http://congreso.seguridad.unam.mx
[2] G. Wolf. "Fortalecimiento del llavero de confianza en un proyecto geográficamente distribuido". http://gwolf.org/node/4055
[3] R. Morris & K. Thompson. "Password Security: A Case History". Communications of the ACM, November 1979. http://www.cs.yale.edu/homes/arvind/cs422/doc/unix-sec.pdf
[4] "Changes in the TLS certificate ecosystem", LWN.net. https://lwn.net/Articles/663875
[5] "Analysis of the strong set in the PGP web of trust". http://pgp.cs.uu.nl/plot/