Capítulo 1

La autenticación en Linux

Primeros pasos

Capítulo 1 La autenticación en Linux

Un requerimiento básico de un sistema informático que realice una actividad importante es la seguridad con la que lleva a cabo sus acciones. Esta seguridad será implementada mediante mecanismos que deben ser adecuados a la información que se pretende proteger. Este conjunto de mecanismos al que nos referimos debe incluir al menos un sistema que se encargue de identificar a las entidades (generalmente usuarios, aunque podrían ser otras aplicaciones) de dicho sistema. El mecanismo del que hablamos se conoce como autenticación y será desarrollado brevemente a lo largo de este capítulo.

1.1 Conceptos básicos

Los sistemas que normalmente usamos las personas para identificar a otras como el aspecto físico o el habla son demasiados complejos para un computador. Sin embargo un computador no tiene que identificarnos sino que tiene que autenticarnos: establecer que un usuario es realmente quien dice ser. Para ello puede realizar diversos métodos de autenticación, que van desde los más simples como introducir una clave secreta hasta el reconocimiento de patrones oculares, por ejemplo.

Los métodos de autenticación pueden clasificarse en tres grupos, en función de lo que utilizan para verificar la identidad: (a) algo que el usuario sabe, (b) algo que éste posee y (c) una característica física del usuario. Este último grupo se conoce como autenticación biométrica.

Cualquier sistema de autenticación debe reunir ciertas características para ser viable; obviamente debe ser fiable con una probabilidad muy elevada (al menos una tasa de error de 10^{-4}); económicamente factible para la entidad (su precio no puede ser superior a lo que se está protegiendo), y ha de soportar con éxito un cierto número de ataques. Por último la característica más importante y que no hemos nombrado es ser aceptado por los usuarios. Un ejemplo típico de este caso sería un sistema basado en análisis de sangre. Aún siendo barato y confiable pocos darían un poco de sangre cada vez que deseen ver el correo.

1.2 Métodos de autenticación

Ya hemos diferenciado tres categorías dentro de los métodos de autenticación. Ahora vamos a verlos un poco más detalladamente.

1.2.1 Sistemas basados en algo conocido

Este es el modelo más básico de autenticación. En él se decide que un usuario es quien dice ser si conoce una palabra o serie de caracteres, que se supone que sólo debe conocer él. Por supuesto esta palabra debe mantenerla en secreto sino el sistema perdería toda fuerza. Cualquiera que la conozca podría suplantar su identidad.

A pesar de los inconvenientes que acabamos de expresar, este modelo es ampliamente utilizado. Es la aproximación más barata y para combatir sus debilidades a menudo se emplea de forma conjunta con otros modelos, como por ejemplo podría ser una tarjeta de crédito y su número pin.

En esencia este es el esquema que se utiliza en Linux aunque de forma más compleja y con substanciales mejoras. Lo veremos con detalle en un apartado posterior.

1.2.2 Sistemas basados en algo poseido

Este esquema basa su funcionamiento en la posesión por parte del usuario de un objeto único. Evidentemente objetos únicos existen pocos así que se llegará a un compromiso de forma que sea difícil de copiar. En esta categoría podemos incluir desde una llave para una cerradura común y corriente hasta una tarjeta inteligente con alta encriptación.

Para acceder a un sistema que use esta autenticación debemos estar registrados en él, esto es, que el sistema reconozca nuestra identidad, y

poseer dicha tarjeta inteligente. El sistema y la tarjeta dialogan según un esquema preestablecido, que puede ser más o menos complejo, pero siempre usando métodos criptográficos de forma que esa información no se pueda capturar ni adulterar.

1.2.3 Sistemas de autenticación biométrica

Existen sistemas donde no se aplica de forma tan necesaria la criptografía sino que establecen sus criterios de acceso mediante el reconocimiento de una característica física humana.

Desde la huella dactilar hasta el análisis de retina existen numerosos métodos, cuyas ventajas son conocidas desde hace largo tiempo: la alta fiabilidad y dificultad de falsificar. Esta última ventaja viene a desmentir la falsa creencia (avivada por multitud de películas de espionaje) de que engañar a un autenticador biométrico es algo fácil. Al contrario, éstos permiten detectar incluso que el patrón utilizado (un dedo o un iris, por ejemplo) pertenecen a un usuario vivo o no.

Las razones por las que estos sistemas no están altamente difundidos es su coste; imposible de asumir para una pequeña entidad, y su dificultad de mantenimiento.

1.3 La autenticación de usuarios en Linux

1.3.1 Autenticación clásica

Hace unos años en un sistema típico Linux, cada usuario poseía un nombre de entrada al sistema o *login* y una clave o *password*. Estos datos se almacenan normalmente en el fichero /etc/passwd, que dispone de una línea para cada usuario. Además en cada línea se guarda diversa información, necesaria para que los usuarios se puedan conectar al sistema y trabajar en él. Veamos una entrada como ejemplo:

jmartin:1ucd4srUn$SjCPJD:500:500:JoseA:/home/jmartin:/bin/bash

Vemos que los campos están separados por el carácter dos puntos (:). El primero de ellos es el login, el nombre de usuario. Así será conocido el usuario en el sistema. El siguiente es la clave. No se encuentra visible en texto plano sino lo que aparece ahí es un hash[1].

Los dos números siguientes corresponden al identificador de usuario (UID) y de grupo (GID), respectivamente. El siguiente campo se conoce como GECOS. Se trata simplemente de información administrativa sobre el usuario, es decir, su nombre real o algún otro tipo de información sobre él.

Los dos últimos campos corresponden al directorio home del usuario, que normalmente será /home/usuario, y el shell con el que comienza por defecto el usuario.

Linux distingue un usuario de otro mediante el UID. El login se realiza con el nombre de usuario básicamente para comodidad de éstos. Evidentemente es más fácil acordarse

1 El Hash es el resultado de aplicar una función resumen a un texto, en este caso al password. Su principal característica es la imposibilidad de llegar a averiguar el texto a partir del hash.

de un nombre que de un número. Así pues si dos usuarios tienen el mismo UID, aunque tengan distintos login y passwords, serán tratados como el mismo, es decir, disfrutarán de los mismos privilegios.

Esta vía puede ser aprovechada por un atacante que introduzca un usuario con UID 0, esto es, de superusuario, y así obtener los permisos de estos.

El fichero /etc/passwd también contiene otras entradas que no corresponden a usuarios reales sino que son utilizadas por ciertos programas, o se mantienen por compatibilidad con otros sistemas. Ejemplos de estas entradas son lp, bin, daemon...

Para evitar cualquier tipo de problema de suplantación, estas cuentas deben estar bloqueadas. Las podemos bloquear mediante el comando passwd y en ese caso en el fichero /etc/passwd aparecerán dos signos de admiración en lugar de la clave (!!).

Para cifrar las claves de acceso de los usuarios Linux utiliza algunas funciones criptográficas. Una de las primeras que se usó fue la función estándar de C crypt(3), basada en el algoritmo DES. La función es irreversible de modo que a partir de la clave cifrada no se puede obtener la clave en claro.

La autenticación del usuario se realiza entonces mediante la aplicación del algoritmo a la clave que éste presenta. El resultado se compara con la contraseña cifrada y si coincide el usuario está autenticado.

De esta forma tan sencilla acceden los usuarios al sistema. Evidentemente podemos poner muchas trabas a este método. A continuación veremos como se ha ido mejorando poco a poco esta autenticación.

1.3.2 Mejora de la seguridad

1.3.2.1 Problemas del modelo clásico

La principal amenaza de un sistema Unix clásico es un ataque de texto cifrado escogido. Resulta muy fácil al atacante tomar claves, cifrarlos y compararlos con las cadenas cifradas que se encuentran en el fichero de claves /etc/passwd. Para remediar un poco esta situación se crearon unas pequeñas marcas, conocidas en inglés como salt. Estas marcas no son más que un número (tradicionalmente desde el 0 al 4095) que se le añadía a la clave de forma que al cifrar dicho conjunto diera como resultado un clave cifrada más larga y por tanto, más difícil de encontrar.

Esta solución, que aunque a priori parezca buena, no lo es tanto. Estamos hablando de ataques de texto escogido. La introducción de la Salt sólo le supone al atacante la inclusión de esos caracteres extras y gracias a la tecnología actual y la existencia de programas como John the Ripper o crack, la ruptura de la clave se puede conseguir en muy poco tiempo. Estos programas utilizan unos archivos de texto, que se conocen como diccionarios, que almacenan cantidades ingentes de claves comunes. Incluyen palabras de diversos ámbitos (cultura, sociedad, familia...) con el fin de abarcar a todos los usuarios.

Estos programas te permiten incluso definir ciertas operaciones para realizar con las palabras del diccionario (invertirlas, añadir números,combinarlas con otras...) de forma que el ataque, con este amplio horizonte, sea más completo y llegue a conseguir su objetivo.

1.3.2.2 Contraseñas aceptables

La principal forma de prevenir los ataques de diccionario es la elección de una buena contraseña, es decir, elegir una palabra que no aparezca en ninguno de estos diccionarios. Por ello, una buena contraseña sería una

combinación de letras (mayúsculas y minúsculas), números y caracteres no alfanuméricos (como pueden ser & ó $).

Para crear una buena contraseña podemos tener en cuenta una serie de reglas a la hora de elegirla:

- No debe contener el nombre de usuario o el real.
- No debe ser una palabra simple (p.ej. Caballo)
- No debe ser una combinación fácil (p.ej. Manuel91)
- Cuanto más caracteres posea, mejor.

Existen diferentes herramientas, como pueden ser Npasswd o passwd+, que le permiten al administrador comprobar lo fuerte que son las contraseñas de sus usuarios. De esta forma puede contactar con ellos para que las cambien y elijan una que se adapte a la política de claves fuertes que lleva el sistema administrado.

Otra opción bastante más cómoda (para el administrador) es el uso de ciertos módulos a la hora de cambiar la contraseña en el sistema. Estos módulos son llamados cuando ejecutamos passwd y tecleamos la nueva clave. Es entonces cuando actúan; tratan de romper la clave según los métodos que hemos mencionado antes y sino pueden la aceptan como válida. Además comprueban las recomendaciones que hemos listado antes (número mínimo de caracteres, combinaciones usuales...).

En particular, nuestro sistema va a utilizar uno de estos módulos, el pam_cracklib.so. Es un módulo PAM. Realiza las comprobaciones que hemos mencionado antes de una forma sencilla y cómoda tanto para el usuario como para el administrador. Veremos este módulo junto con el resto de módulos PAM en el siguiente capítulo.

Por último recordar que aunque se sigan todas las instrucciones dadas y se elija una buena clave, pierde toda su robustez si se comparte con alguien. Como curiosidad podemos nombrar un principio que se utiliza en los libros de lengua anglosajona sobre seguridad. Este es el principio KISS (Kepp It Secret Stupid), que viene a significar que el principal riesgo es mostrar o compartir la clave.

1.3.2.3 Shadow Password

Otro método que se viene usando desde hace ya algún tiempo es el oscurecimiento de las contraseñas o Shadow Password. Se trata de que usuarios no privilegiados puedan leer el fichero donde se encuentran las claves cifradas /etc/passwd. Vimos antes que todos los usuarios deben poder leer este fichero para el correcto funcionamiento del sistema. Así que lo que haremos será permitir los accesos en lectura, pero con la diferencia de que la clave cifrada no se encontrará ya en el fichero /etc/passwd (aparecerá un símbolo en su lugar, generalmente una x) sino en el /etc/shadow, que sólo el usuario root puede leer.

jmartin:x:Jose A:/home/jmartin:/bin/bash

En cuanto al fichero /etc/shadow tiene un aspecto muy parecido al de claves. La entrada correspondiente a la línea de arriba sería la siguiente:

jmartin:1ucd4srUn$SjXGJCAPJD:12702:-1:99999:-1:::

Los campos que componen dicha entrada son el nombre de usuario, la clave cifrada y diversa información relacionada con la seguridad de las claves y que veremos a continuación.

En cuanto a la adopción de este mecanismo debemos decir que ha sido ampliamente acogido. Es raro (o casi imposible) encontrar una distribución de Linux que no lo implemente. Las más nuevas incluso no lo incluyen como opción, sino que simplemente se usa por defecto.

Es realmente un gran avance. Desde siempre el acceso de los usuarios a las claves cifradas ha supuesto un gran problema seguridad, ahora zanjado.

1.3.2.4 Envejecimiento de contraseñas

Prácticamente todas las implementaciones de shadow password incluyen otro mecanismo, conocido como envejecimiento de contraseñas (aging password). Se trata de proteger los passwords de los usuarios mediante la limitación de su uso durante un tiempo determinado. Pasado este tiempo la contraseña no será válida y deberá ser cambiada por el usuario.

Este método protege al usuario en dos situaciones básicamente:

a) Un ataque de fuerza bruta.

Supongamos un usuario, que ha seguido las recomendaciones que dimos para la elección de la contraseña, y trabaja con su robusta clave. Por alguna razón su clave cifrada cae en “malas manos”, las cuales ejecutan un programa para conseguir claves (reventar claves, según el argot criptográfico) por fuerza bruta. Pues bien, si la contraseña de este buen usuario no cambiase, la seguridad se vería comprometida. Aunque la fuerza bruta tardase ocho meses, por poner un ejemplo, al final acabaría consiguiéndola y el acceso al sistema estaría comprometido.

Un detalle que siempre se menciona cuando hablamos de fuerza bruta es la longitud de las claves. Se afirma que una clave larga es difícil de romper mediante este sistema y en todo caso se tardaría muchísimo tiempo en

comprobarlas todas. Bien, esto es en teoría y usando la tecnología actual. Tenemos que contemplar la posibilidad de que se pueda averiguar en un momento dado. Además, como la capacidad de procesamiento no deja de crecer en las máquinas más modernas y el uso de clústeres cada vez más grandes no podemos fiarnos de frases como "tardaría 100 años en desencriptarse". Como curiosidad vamos a reproducir una cita:

"Ms-dos es capaz de paginar hasta 1 megabyte de memoria ya que nunca jamás a ningún sistema le hará falta más". Bill Gates, 1979.

Debemos ser previsores y no dejar cabo suelto. Hay que llegar a un compromiso entre la seguridad del sistema y causar molestias a los usuarios. Es tarea del administrador decidir si aplicar este método, y en caso afirmativo determinar el tiempo de envejecimiento.

b) Clave interceptada en una sesión no cifrada.

En este caso el envejecimiento de las contraseñas previene del uso de contraseñas capturadas al usar algún servicio como telnet o ftp.

Cuando usamos estos servicios, cualquier equipo entre el nuestro y el servidor puede leer los paquetes que se envían por la red, incluyendo aquellos que llevan nuestro nombre de usuario y contraseña. Así que de esta forma un atacante situado entre los dos equipos puede obtener nuestros login y password. Si la clave capturada es válida indefinidamente, esa persona tiene un acceso asegurado al servidor en el momento que quiera; sin embargo, si la clave tiene un periodo de vida, el atacante sólo podrá utilizarla antes de que el sistema nos obligue a cambiarla.

A primera vista, puede parecer que la utilidad del envejecimiento de contraseñas no es muy grande; al fin y al cabo, la lectura de paquetes destinados a otros equipos (sniffing) no se hace por casualidad: el atacante

que lea la red en busca de claves y nombres de usuario lo va a hacer porque quiere utilizar estos datos contra un sistema. Sin embargo, una práctica habitual es dejar programas escuchando durante días y grabando la información leída en ficheros; cada cierto tiempo el pirata consultará los resultados de tales programas, y si la clave leída ya ha expirado y su propietario la ha cambiado por otra, el haberla capturado no le servirá de nada a ese atacante.

Los periodos de expiración de las claves se suelen definir a la hora de crear a los usuarios con las herramientas que cada sistema ofrece para ello. Si queremos modificar alguno de estos periodos una vez establecidos, desde esas mismas herramientas de administración podremos hacerlo, y también desde línea de comandos mediante órdenes como chage o usermod. Como hemos dicho antes, en el archivo /etc/shadow se almacena, junto a la clave cifrada de cada usuario, la información necesaria para implementar el envejecimiento de contraseñas; una entrada de este archivo es de la forma

```
jmartin:LEgPN8jqSCHCg:10322:0:99999:7:::
```

Tras el login y el password de cada usuario se guardan los campos siguientes:

- Días transcurridos desde el 1 de enero de 1970 hasta que la clave se cambió por última vez.
- Días que han de transcurrir antes de que el usuario pueda volver a cambiar su contraseña.
- Días tras los cuales se ha de cambiar la clave.
- Días durante los que el usuario será avisado de que su clave va a expirar antes de que ésta lo haga.
- Días que la cuenta estará habilitada tras la expiración de la clave.
- Días desde el 1 de enero de 1970 hasta que la cuenta se deshabilite.
- Campo reservado.

Como podemos ver, cuando un usuario cambia su clave el sistema le impide volverla a cambiar durante un periodo de tiempo; con esto se consigue que cuando el sistema obligue a cambiar la contraseña el usuario no restaure inmediatamente su clave antigua (en este caso el esquema no serviría de nada). Cuando este periodo finaliza, suele existir un intervalo de cambio voluntario: está permitido el cambio de contraseña, aunque no es obligatorio; al finalizar este nuevo periodo, el password ha expirado y ya es obligatorio cambiar la clave. Si el número máximo de días en los que el usuario no puede cambiar su contraseña es mayor que el número de días tras los cuales es obligatorio el cambio, el usuario no puede cambiar nunca su clave. Si tras el periodo de cambio obligatorio el password permanece inalterado, la cuenta se bloquea.

1.3.2.5 Claves de un solo uso

El envejecimiento de contraseñas tiene dos casos extremos. Por un lado, tenemos el esquema clásico: una clave es válida hasta que el usuario voluntariamente decida cambiarla (es decir, no hay caducidad de la contraseña). El extremo contrario del Aging Password es otorgar un tiempo de vida mínimo a cada clave, de forma que sólo sirva para una conexión: es lo que se denomina clave de un solo uso, One Time Password.

¿Cómo usar claves de un solo uso? Podemos ver su uso mediante diferentes aproximaciones; la más simplista consiste en asignar al usuario una lista en papel con la secuencia de claves a utilizar, de forma que cada vez que éste conecte al sistema elimina de la lista la contraseña que acaba de utilizar.

Por su parte, el sistema avanza en su registro para que la próxima vez que el usuario conecte pueda utilizar la siguiente clave. Otra aproximación consiste en utilizar un pequeño dispositivo que el usuario debe llevar consigo, como una tarjeta o una calculadora especial, de forma que cuando

desee conectar el sistema le indicará una secuencia de caracteres a teclear en tal dispositivo; el resultado obtenido será lo que se ha de utilizar como password. Para incrementar la seguridad ante un robo de la tarjeta, antes de teclear el número recibido desde la máquina suele ser necesario utilizar un código PIN que el usuario debe mantener en secreto.