PRÁCTICA 4: INTRODUCCIÓN A LA PROGRAMACIÓN SSL.

Héctor Menéndez

AIDA Research Group
Computer Science Department
Universidad Autónoma de Madrid

17 de abril de 2013

Index

Objetivos

Los objetivos de la presente práctica son:

- Familiarizarse con los mecanismos de seguridad.
- Comprender las ventajas y desventajas de distintos algoritmos de encriptación.
- Implementar en programas la seguridad de acceso de tipo UNIX con `shadows`.
- Implementar la seguridad en comunicaciones usando la capa SSL.

Ejercicio 1: Comprobación de la clave de usuario

- Los ficheros `/etc/passwd` y `/etc/groups` contienen la información acerca de los usuarios incluidas las claves de acceso (legibles).
- La falta de seguridad que implica esta capacidad de lectura forzó a introducir un sistema auxiliar denominado `shadows`.
- `/etc/shadows` es accesible para los administradores del sistema.
- Este ejercicio se deberá probar como administrador del sistema.

Ejercicio 1: Comprobación de la clave de usuario

- El formato del campo que contiene la password es `$en$salt$clave`. Donde *en* puede ser:
  - 1– para la codificación MD5
  - 2a– para Blowfish
  - 5– para SHA–256
  - 6– para SHA–512

Ejercicio 1: Comprobación de la clave de usuario

- Funciones de C y la familia de funciones a las que pertenece: `getpwnam`, `getspnam`, `MD5_Init` o `blowfish`. buscar o implementar las funciones de codificación que necesite aparte de las indicadas.
- Se puede utilizar cualquier librería disponible para implementar las funciones.

Ejercicio 1: Comprobación de la clave de usuario

Dado un usuario y una clave, compruebe que la clave es correcta. Las condiciones de funcionamiento son:

- Nombre comprobación.
- Banderas posibles `-u` (usuario) y `-p` (pass).
- Pueden usarse en cualquier orden.
- Si no se indica usuario se utilizará el usuario con el que ese está ejecutando.
- Si no se indica ninguna clave, se solicitará su introducción (no tiene por qué ser invisible).
- Debe ser capaz de comprobar la clave en las cuatro codificaciones indicadas anteriormente.

Ejercicio 1: Comprobación de la clave de usuario

El alumno deberá presentar una descripción del subsistema de autenticación de UNIX incluyendo el subsistema `shadows`.

Ejercicio 2: Implementación de sockets SSL

- Se partirá de la P1 se implementará una comunicación segura con SSL.
- Como ejemplos se pueden usar los siguientes servidores:
  - `eu.geekshed.net:+6697`
  - `irc.dal.net:+6697`
  - `irc.distributed.net:+994`
  - `irc.freenode.net:+7000`
  - `irc.indymedia.org:+6697`
  - `irc.link-net.org:+7000`
  - `irc.oftc.net:+6697`
  - `irc.rizon.net:+6697`
  - `irc.theonering.net:+6697`
  - `ssl.efnet.org:+9999`

Ejercicio 2: Implementación de sockets SSL

Para iniciar la comunicación segura es necesario realizar los siguientes pasos:

1. Establecer una conexión TCP normal.
2. Registrar las cadenas de error para libcrypto y libssl (librerías que hay que utilizar a la hora de compilar).
3. Registrar los cifrados disponibles.
4. Creación de un contexto en el que se indica si se es cliente o servidor y si se utiliza SSL–2 o SSL–3.
5. Creación de una estructura de conexión SSL.
6. Registrar la estructura SSL creada a la conexión TCP inicialmente creada.
7. Iniciar el *handshake* en el que se realiza el intercambio de información para negociar la comunicación y se intercambian las claves necesarias.

Ejercicio 2: Implementación de sockets SSL

Para finalizar las conexión segura es necesario:

1. Desconectar el SSL.
2. Liberar la estructura de conexión SSL.
3. Liberar el contexto de la conexión.

Ejercicio 2: Implementación de sockets SSL

- La lectura y la escritura también deben realizarse de forma codificada.
- Para realizar estas tareas se puede utilizar OpenSSL.
- Se facilita un código de ejemplo de un módulo de sockets de OpenSSL

Memoria del ejercicio

La memoria debe incluir:

- La introducción.
- Un manual completo de los programas solicitados.
- Un apartado de pruebas para cada uno de los ejercicios.
- Un apartado de desarrollo técnico de cada uno de los ejercicios.
- Un apartado explicando el subsistema de autenticación de UNIX incluido shadows.
- Un apartado explicando cómo se ha comprobado que la comunicación segura es verdaderamente segura.

Imprescindibles

Se considerará como imprescindible en esta práctica:

- Generar adecuadamente las claves en MD5, SHA–256 y SHA–512.
- Realizar una conexión SSL correcta con un control de errores correcto.
- Realizar el envío y la recepción de mensajes codificados en SSL de forma correcta.

Imprescindibles

- Una memoria que posea:
  - Un manual de uso del software.
  - Una descripción técnica del proyecto.
  - Un conjunto de pruebas cuyos resultados estén bien explicados.
  - Una descripción del subsistema de seguridad en UNIX incluido shadows.
  - Si se han utilizado librerías distintas de las incluidas en un UNIX con SSL es necesario indicar qué librerías se están utilizando y hacer una breve descripción de estas.
  - Un conjunto de pruebas suficientemente completo y la explicación de dichas pruebas.