5.PRELUDE-IDS.

5.1.Introducción.

Para la implementación del colector de eventos usaremos Prelude 0.9.0. A continuación se detalla la instalación de dicho software.

5.2.Instalación de Prelude

5.2.1.Instalación de los paquetes en Gentoo Linux.

En primer hay que comprobar que en la base de datos de paquetes de gentoo (online) la versión de libprelude-0.9.0, libpreludedb-0.9.0, prelude-manager-0.9.1 y demás paquetes para nuestra arquitectura hardware (x86 en nuestro caso = Intel) es estable. Si aparece como Masked (en rojo) hay que hacer lo siguiente:

Editamos (o creamos) el archivo /etc/portage/package.keywords escribiendo las siguientes líneas:

```
dev-libs/libprelude ~x86
dev-libs/libpreludedb ~x86
app-admin/prelude-lml ~x86
app-admin/prelude-manager ~x86
net-analyzer/prewikka ~x86
```

A continuación ya podemos instalar con emerge el paquete Prelude. Hay que saber que para la base de datos de eventos, si no disponemos de ningún lenguaje instalado, es muy conveniente usar PostgreSQL por su flexibilidad y por tener licencia libre.

```
# emerge -av dev-db/postgresql
```

Con Prelude-manager se instalará como librerías dependientes libprelude y libpreludedb.

```
# USE="dbx xml" ACCEPT_KEYWORDS="~x86" emerge =prelude-manager-0.9.1
```

Prewikka es el analizador de red, y snort para los sensores. La herramienta Oinkmaster permitirá actualizar las reglas que definen los ataques IDS, por lo que será fundamental su instalación conjunta con la del IDS Snort.

```
# ACCEPT_KEYWORDS=”~x86” emerge prelude-lml-0.9.1.ebuild
# ACCEPT_KEYWORDS="~x86" emerge prewikka prelude-lml
# USE="postgres prelude ssl" ACCEPT_KEYWORDS="~x86" emerge snort
  oinkmaster
```

Por último, instalamos los paquetes nagios y nessus. Nagios es una herramienta para chequeo de logs y máquinas y será de mucha utilidad en el colector de eventos. Snort será fundamental para que los sensores emitan logs IDS.

```
# emerge prelude-nessus prelude-nagios
```

Si no está instalado apache, la instalación de nessus y nagios dará un error. Para solventarlo, apache deberá ser instalado:

```
USE="threads apache2" emerge apache
```

5.2.2.Instalación en resto de Linux.

5.2.2.1. Paquetes necesarios para Prelude.

Descargamos los binarios de las páginas de Prelude (http://www.prelude-ids.com/download/releases/) y PostgreSQL (http://www.postgresql.org/ftp/source/v8.1.3/). Los binarios descargados para una instalación de ejemplo en Linux OpenSuse 10.0 han sido los siguientes:

```
linux~# ls
     postgresql-8.1.3.tar.bz2          libprelude-0.9.8.tar.gz
     libpreludedb-0.9.7.1.tar.gz       prelude-lml-0.9.4.tar.gz
     prelude-manager-0.9.4.1.tar.gz    gnutls-1.2.10.tar.bz2
     libgcrypt-1.2.2.tar.gz            libgpg-error-1.3.tar.bz2
linux~# tar -xvjf postgresql-8.1.3.tar.bz2
```

Para instalar postgresql es necesario tener en el $PATH la ruta al compilador gcc o cc. Teniendo esto basta con teclear lo siguiente:

```
linux~/postgresql-8.1.3# ./configure --datadir=/var/lib/postgres/data --with-perl
  --prefix=/usr --without-readline --without-zlib
linux~/postgresql-8.1.3# make && make install
```

A continuación instalaremos en primer lugar las librerías de prelude. Dichas librerías necesitan de las **libgnutls** las cuales necesitan de **libgcrypt** ([17])**.** Ésta última tiene una dependencia de instalación con la librería **libgpg-error**. Si no disponemos de ninguna de estas librerías, es necesario descargarlas de los sitios ftp que se detallan a continuación, por ejemplo con *wget*:([18])

```
linux~# wget http://ftp.gnupg.org/gcrypt/libgpg-error/libgpg-error-1.3.tar.bz2
linux~# wget ftp://ftp.gnupg.org/gcrypt/libgcrypt/libgcrypt-1.2.2.tar.gz
linux~# wget ftp://gnutls.hellug.gr/pub/gnutls/gnutls-1.2.10.tar.bz2
linux~# wget ftp://xmlsoft.org/libxml/libxml2-2.6.11.tar.gz
linux~# firefox http://sourceforge.net/projects/libidmef
```

Tras descomprimir los paquetes ejecutamos el guión de configuración para cada una de las librerías con los parámetros siguientes:

```
linux~/libgpg-error-1.3# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --build=i686-pc-linux-gnu
linux~/libgcrypt-1.2.2# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --enable-noexecstack
  --disable-dependency-tracking --with-pic --build=i686-pc-linux-gnu
linux~/gnutls-1.2.10# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --without-included-minilzo
  --without-included-libtasn1 --without-included-opencdk --enable-gtk-doc
  --build=i686-pc-linux-gnu
```

Si el guión de configuración no encuentra la ruta de alguna de las librerías, hay que incluir en la ruta de librerias el directorio /usr ó /usr/lib. Dicha ruta LD_LIBRARY_PATH se especifica como líneas del fichero ***/etc/ld.so.conf***, en el caso de Debian por ejemplo.

[17] Por ejemplo, la librería gnutls-1.2.10 requiere una versión mayor o igual a la libgcrypt-1.2.2, que es precisamente la versión instalada en este caso.

[18] Dichos servidores ftp se pueden incluir en ficheros como /etc/make.conf para **emerge** de gentoo o /etc/apt/sources.list para el **apt-get install** de debian.

5.2.2.2. Instalación de Prelude-IDS.

Lo siguiente es proceder a la instalación de prelude. Se comienza por la librería libprelude, que deberá ir en el directorio /usr y se activará para los sensores perl (y python de forma opcional)

```
linux~/libprelude-0.9.8# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --enable-perl --enable-python
  --build=i686-pc-linux-gnu
```

Para el módulo siguiente es impreescindible que se instale con el soporte para Postgresql. Este paquete instalará en /usr/share/libpreludedb/ los archivos ".sql" para la generación de las tablas de la base de datos.

```
linux~/libpreludedb-0.9.7.1# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --enable-gtk-doc --enable-mysql=no
  --enable-pgsql --enable-perl --enable-python --build=i686-pc-linux-gnu
```

Para la instalación del gestor Prelude-Manager se puede habilitar el que los mensajes se escriban en un fichero en XML y/o modo texto.

```
linux~/libxml2-2.6.11# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --build=i686-pc-linux-gnu
linux~/prelude-manager-0.9.4.1# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --enable-gtk-doc --localstatedir=/var
  --build=i686-pc-linux-gnu
```

Por último, para los sensores que no tengan soporte para transmitir con formato IDMEF, se instala Prelude-LML. Las reglas para rellenar los campos se definen con reglas regulares por lo que tendremos que tener ***pcre***, acrónimo de Perl Common Regular Expressions ([19]).

```
linux# wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-6.6.tar.bz2
linux~/pcre-6.6# ./configure --prefix=/usr --host=i686-pc-linux-gnu --enable-utf8
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib –build=i686-pc-linux-gnu
linux~/prelude-lml-0.9.4# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --enable-gtk-doc
  --build=i686-pc-linux-gnu
```

Otra librería útil para que otro tipo de aplicaciones usen IDMEF es libidmef. Esto será de especial interés en un futuro que desarrollemos aplicaciones con soporte para IDMEF de Prelude. Para su instalación, el guión de configuración debería ser tal que así:

```
linux~/libidmef-1.0.2# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib –build=i686-pc-linux-gnu
```

[19] En Linux Gentoo es **libpcre**.

5.2.2.3. Instalación y configuración de Snort-IDS con MarteGUI ®.

Por último descargamos e instalamos el sistema de detección de intrusiones Snort. Las reglas de snort, parches etc se pueden descargar automáticamente desde los servidores de Gentoo o por medio de la interfaz MarteGUI ([20]).

```
linux~# wget www.snort.org/dl/current/snort-2.4.4.tar.gz
linux~/snort-2.4.4# ./configure --prefix=/usr --host=i686-pc-linux-gnu
  --mandir=/usr/share/man --infodir=/usr/share/info --datadir=/usr/share
  --sysconfdir=/etc --localstatedir=/var/lib --with-postgresql --without-mysql
  --with-openssl --without-odbc --without-oracle --enable-prelude
  --without-sguil--build=i686-pc-linux-gnu
```

Si queremos activar snort-inline, hay que instalar libpcap (www.tcpdump.org). ([21])

OpenSSL se puede activar de forma opcional descargando el paquete de www.openssl.org/source

Comandos necesarios para la configuración y descarga de reglas de Snort desde MarteGUI.

```
/sbin/linkether
/sbin/ip
/sbin/sudo -> /usr/bin/sudo
```

5.2.3.Configuración del Gestor de Prelude (Prelude-Manager).

Es necesario que editemos el fichero de configuración

Para nuestro caso vamos a usar que el puerto de escucha de Prelude-Manager será el 4960 y el de la base de datos de eventos el 3306.

/etc/prelude-manager/prelude-manager.conf

```
listen = 127.0.0.1:4960

[db]
type = pgsql
host = localhost
port = 5432
```

[20] Desde los servidores Gentoo, los paquetes se encuentran en: http://gentoo.gg3.net/distfiles/snortrules-pr-2.4.tar.gz, http://gentoo.gg3.net/distfiles/Community-Rules.tar.gz http://gentoo.gg3.net/distfiles/snort-2.4.0-genpatches.tar.bz2

[21] Al intentar hacer el .configure del snort, si dice que no encuentra "libipq.h", es que no tenemos instalado iptables. Esta librería tiene por nombre las iniciales del acrónimo "iptables userspace packet queuing library". Por tanto, descargamos la última versión de iptables. Esto sólo si queremos activar SNORT-INLINE.

Descarga de las libpcap: http://gentoo.gg3.net/distfiles/libpcap-0.9.4.tar.gz

```
libpcap# ./configure --prefix=/usr --host=i686-pc-linux-gnu --mandir=/usr/share/man --
   infodir=/usr/share/info --datadir=/usr/share --sysconfdir=/etc --localstatedir=/var/lib --
   enable-ipv6 -build=i686-pc-linux-gnu
```

Descarga de iptables: http://www.netfilter.org/projects/iptables/files/iptables-1.3.5.tar.bz2

```
iptables# make KERNEL_DIR=/usr/src/linux BINDIR=/usr/bin LIBDIR=/usr/lib MANDIR=/usr/man

iptables# make KERNEL_DIR=/usr/src/linux BINDIR=/usr/bin LIBDIR=/usr/lib MANDIR=/usr/man
install
```

```
name = eventdb
user = preluderoot
pass = proyecto
```

Dentro del mismo fichero, podemos configurar las propiedades de XMLMod y TextMod

```
[TextMod]
logfile = stderr
logfile = /var/log/prelude.log

[XmlMod]
disable-buffering
validate
format
logfile = stderr
logfile = /var/log/prelude-xml.log
```

5.2.4.Creación de la base de datos de Prelude con PostgreSQL.

5.2.4.1. Inicio con Gentoo Linux.

Tras la descarga del paquete postgresql-8.0.3, necesitamos que se esté ejecutando el servicio postgresql, para lo cual hay que crear un lugar para almacenar la base de datos y su configuración. Es muy importante cambiar el número de versión de PostgreSQL en el siguiente comando, si es necesario ([22]).

```
# emerge --config =postgresql-8.0.3
```

Arrancamos el servidor PostgreSQL con las opciones por defecto porque es necesario que la creación de los usuarios se haga conectándose a la base de datos *template1* por un socket_Unix y puerto 5432. Esto se consigue con el siguiente comando:

```
# /etc/init.d/postgresql start
```

Si tenemos instalada otra distribución de linux, o si se desea arrancar el servidor PostgreSQL de otro modo diferente con idénticos resultados, debemos teclear:

```
# su - postgres
postgres# postmaster -D /var/lib/postgresql/data/
LOG:  el sistema de bases de datos fue apagado en 2005-11-21 18:54:11 CET
LOG:  el registro de checkpoint está en 0/A32A30
LOG:  registro de redo en 0/A32A30; registro de undo en 0/0; apagado TRUE
LOG:  siguiente ID de transacción: 544; siguiente OID: 17230
LOG:  el sistema de bases de datos está listo
```

Si no ejecutamos el *postmaster* en background, deberemos cambiar a otra ventana de shell porque en esa irán saliendo los mensajes de registro satisfactorios y de error.

5.2.4.2. Inicio genérico con Linux.

Una vez instalado el paquete tgz (comprimido en tar y gzip) nos autenticamos como root, luego como el usuario postgres e inicializamos la base de datos en el directorio de datos que especifiquemos. Para este caso ha sido /var/lib/postgresql/data

```
postgres# initdb -D /var/lib/postgresql/data
postgres# postmaster -D /var/lib/postgresql/data &
```

[22] `emerge --config =postgresql ↵` también es válido sin especificar versión.

Con el último comando se arranca el servicio postmaster en el puerto por defecto de escucha 5432.

5.2.4.3. Creación del base de datos y root PostgreSQL.

Configuramos que el servidor de PostgreSQL escuche en el puerto deseado, ya que el puerto por defecto sería 5432. Esto es recomendable (fichero de configuración de PostgreSQL) y suficiente con especificar el puerto, pudiendo dejar la dirección de escucha en default, que es lo mismo que no especificarlo comentándo la línea correspondiente.

/var/lib/postgresql/data/postgresql.conf

```
#listen_addresses = 'localhost'
port = 3306
max_connections = 100
```

No obstante, el usuario creador de bases de datos *root* debe haberse creado cuando el postmaster escucha del puerto por defecto 5432. Por tanto, aprovechamos para crear el usuario *root* antes de reiniciar el servicio postgresql.

La opción -W sirve para dotar de contraseña al usuario root.([23])

```
# su - postgres
postgres# createuser -W preluderoot -d -s
Contraseña: prelude
CREATE ROLE
```

Le permitiremos al root de la base de datos crear nuevos usuarios con **-s** y bases de datos con **-d.** Luego escribimos una contraseña.

También como usuario *postgres* de nuestra máquina, creamos la base de datos, por ejemplo, de nombre "eventdb".

```
postgres# createdb -W eventdb
Contraseña: prelude
CREATE DATABASE
```

Estos nombres de la base de datos y la contraseña deben concordar con los del fichero /etc/prelude-manager/prelude-manager.conf.

Otra forma de hacerlo es mediante comandos SQL, accediendo previamente a la base de datos con el comando siguiente:

```
postgres# psql eventdb -U preluderoot
```

Entramos en la consola de la base de datos y creamos el usuario preludeuser, el cual podrá tener acceso total a dicha base de datos, aunque no podrá crear usuarios ni bases de datos. Esto también se podría haber hecho desde la consola como root del systema utilizando el comando *createuser -W preludeuser*, tal como se usó anteriormente para root del PostgreSQL.

```
eventdb=# CREATE USER preludeuser WITH ENCRYPTED PASSWORD 'prelude'
          NOCREATEDB NOCREATEUSER;
CREATE ROLE
```

Las versiones de la serie 8.0.x de PostgreSQL permiten que coexistan un usuario y una base de datos con el mismo nombre, ya que se diferencia entre CREATE USER y CREATE DATABASE. A partir de la serie 8.1.x, ambos entes serán CREATE ROLE.

[23] Para ello puede verse /usr/bin/createuser --help

5.2.4.4. Actualización del esquema de la base de datos de prelude.

Como tenemos ya instaladas las librerías preludedb en /usr, el paso final es crear las tablas y actualizarlas. Los siguientes comandos los ejecutamos tras salir con \q

```
postgres# psql -U preludeuser eventdb
Bienvenido a psql 8.1.2, el terminal interactivo de PostgreSQL.

Digite:  \copyright para ver los términos de distribución
         \h para obtener ayuda sobre comandos SQL
         \? para obtener ayuda sobre comandos internos
         \g o punto y coma (;) para ejecutar consulta
         \q para salir

eventdb=> \i /usr/share/libpreludedb/classic/pgsql.sql
eventdb=> \i /usr/share/libpreludedb/classic/pgsql-update-14-3.sql
eventdb=> \i /home/aledavid/table_protocol.sql
eventdb=> \i /home/aledavid/table_service.sql
```

Ojo, el anterior comando no se da desde el usuario root del sistema, sino desde el postgres. Pueden ser necesarias otro tipo de actualizaciones si se utiliza una versión def prelude-manager posterior a la 0.9.1. Para ello se debería inspeccionar el directorio /usr/share/libpreludedb/classic/ y actualizar con \i y el fichero más nuevo de pgsql-update.

5.2.4.5. Rearranque del Postmaster con el puerto deseado.

Para la distribución Gentoo Linux basta con rearrancar el servicio postgresql, ya que el archivo de configuración correspondiente ha sido convenientemente editado.

```
usuario@localhost# su -
# /etc/init.d/postgresql start
# rc-update add postgresql default
```

Si no disponemos de Gentoo Linux, proponemos otra forma alternativa. Arrancamos el servidor postmaster y ejecutamos el servicio postgresql desde root. Para no tener dos servicios postmaster ejecutándose, hacemos lo siguiente:

```
# kill -9 `pidof postmaster`
```

Si hay algún error con el postmaster.pid al rearrancar el servicio *postgresql*, podemos especificar manualmente las opciones del fichero de configuración de postgre en la línea de comandos ejecutando postmaster:

```
postgres# postmaster -i -D /var/lib/postgresql/data/ -p 3306
                                                    -h localhost >logfile 2>&1 &
```

-i sirve para habilitar conexiones TCP/IP.

-D especifica el directorio en el que buscará los archivos postmaster.

-p especifica el puerto de conexión (3306 es el que usará la base de datos prelude).

-N 100 será el número máximo de conexiones.

-h localhost, donde localhost es el nombre de la máquina en la que se ejecuta el server.

El resto de parámetros es para almacenar los mensajes en un fichero de logs y para ejecutar dicha acción en background.

5.2.5.Conexión de prelude-manager a la base de datos de eventos.

5.2.5.1. Creación de perfil prelude–manager.

Para probar el entorno de trabajo creamos el usuario prelude manager desde la shell y como usuario root, lo cual generará el siguiente código:

```
# prelude-adduser add prelude-manager --uid 0 --gid 0

- Using default TLS settings from /usr/local/etc/prelude/default/tls.conf:
  - Generated key size: 1024 bits.
  - Authority certificate lifetime: unlimited.
  - Generated certificate lifetime: unlimited.

- Creating analyzer prelude-manager.
  - Creating /usr/local/etc/prelude/profile/prelude-manager...
  - Allocated ident for prelude-manager: 1318899690433525.
    - Generating RSA private key... This might take a very long time.
      [Increasing system activity will speed-up the process.]

    - Generating 1024 bits RSA private key... Done.

  - Creating /usr/local/var/spool/prelude/prelude-manager...
```

Pretendemos ejecutar prelude-manager, para que se suscriba a la base de datos de eventos y a los ficheros de TextMod y xmlMod antes de comenzar el servicio prelude-manager.

Para ello debemos cerciorarnos que el servicio postgresql tiene habilitado el puerto tcp de escucha 3306 que configuramos previamente en /etc/prelude-manager/prelude-manager.conf, para lo cual ejecutamos el comando:([24])

```
# netstat --inet -apn
Active Internet connections (servers and established)
Proto ... Local Address  Foreign Address  State   PID/Prog
tcp   ... 127.0.0.1:611   0.0.0.0:*       LISTEN  8915/famd
tcp   ... 127.0.0.1:3306  0.0.0.0:*       LISTEN  11741/postmaster
tcp   ... 0.0.0.0:111     0.0.0.0:*       LISTEN  8902/portmap
```

Con esto, debe aparecer como dirección Ipv4 ó Ipv6 el servicio *postgres* y estado LISTEN

Si no aparece, lo mejor es cerciorarse de que el servidor está funcionando. Por tanto lo configuramos para que acepte conexión TCP/IP, ya que por defecto buscará Unix_sockets. Ejecutamos en background el postmaster, ya que como root no se permite por seguridad.

```
postgres# postmaster -i -D /var/lib/postgresql/data -p 3306 -N 100
                     -h localhost >logfile 2>&1 &
```

Si no vamos a escuchar desde localhost sino desde otra IP, debemos añadir dicha dirección al fichero siguiente:

/var/lib/postgresql/data/pg_hba.conf

```
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
host    all         all         127.0.0.1/32          trust
hostnossl    all         all         192.168.100.72/24     trust
```

[24] #netstat --inet6 -apn para IPv6.

Tras la configuración del host-no-ssl, hay que tener deshabilitado el ssl=off en el postgresql.conf y reiniciar el servicio postmaster.

A continuación ejecutamos el gestor de prelude, desde una shell como root. La ejecución en background es opcional.

```
# prelude-manager &
[1] 10193
# - server started (listening on 127.0.0.1 port 4690).
- Subscribing db[default] to active reporting plugins.
- Subscribing TextMod[default] to active reporting plugins.
- Subscribing XmlMod[default] to active reporting plugins.
- Generating 1024 bits Diffie-Hellman key for TLS...
```

5.2.6.Errores comunes.

5.2.6.1. Al crear/acceder a la base de datos eventdb o al crear root de Postgre.

```
could not connect to server: No existe el fichero o el directorio
        Is the server running locally and accepting
        connections on Unix domain socket "/tmp/.s.PGSQL.5432"?
```

Este error se da cuando queremos crear una base de datos, acceder a ella y/o crear un usuario de la misma corriendo postmaster sin que escuche del puerto por defecto 5432. Se soluciona cambiando el /var/lib/postgresql/data/postgresql.conf especificando el puerto 5432 y rearrancando el servicio, o ejecutando el siguiente comando:

```
postgres# ln /tmp/.s.PGSQL.3306 /tmp/.s.PGSQL.5432
```

5.2.6.2. Al arrancar prelude-manager

```
server started (listening on 127.0.0.1 port 4960)
Option error: could not initialize libpreludedb: Database schema version too old.
```

Indica que después de haber accedido a la base de datos con el comando *psql* hemos actualizado la base de datos incorrectamente, usando el archivo ".sql" inapropiado. Se soluciona ejecutando

```
eventdb=> \i /usr/share/libpreludedb/classic/pgsql-update-14-3.sql
```

o también:

```
# PGPASSWORD=proyecto psql -d eventdb -U preluderoot
                    < /usr/share/libprelude/classic/pgsql-update-14-3.sql
```

5.2.6.3. Error al conectar prelude-manager a un host distinto del host por defecto (localhost)

```
palocortado aledavid # prelude-manager &
[1] 2214
- server started (listening on 192.168.100.72 port 4690).
Option error: could not initialize libpreludedb: Connection error: could not
 connect to server: Connection refused
        Is the server running on host "192.168.100.72" and accepting
        TCP/IP connections on port 5432?.
```

5.2.6.4. Error de acceso a Prelude-Manager.

```
- Subscribing db[default] to active reporting plugins.
- Subscribing TextMod[default] to active reporting plugins.
- Subscribing XmlMod[default] to active reporting plugins.
  No es posible acceder a .../spool/prelude-manager
```

Se debe a que el sistema no permite el algoritmo de encriptación de clave privada que usa prelude-manager-0.9.1. Se soluciona instalando los módulos apropiados de TLS y RSA o volviendo a la versión prelude-manager-0.9.0, que sí soporta el algoritmo Diffie-Hellman.

5.2.6.5. Error *Invalid section [db]* en prelude-manager.conf.

```
- Subscribing db[default] to active reporting plugins.
  Invalid section [db]
```

Se debe a que no están instaladas correctamente las libpreludedb o no está actualizada la versión.

5.2.7. Actualización rc-update y esquema general del colector.

El arranque del gestor de prelude también se podría haber hecho con el siguiente comando:

```
# /etc/init.d/prelude-manager start
```

Por último, añadimos al grupo por defecto de rc para que se cargue la próxima vez en el arranque.

```
# rc-update add prelude-manager default
```

El esquema entre la base de datos de eventos y el gestor de prelude es el siguiente:

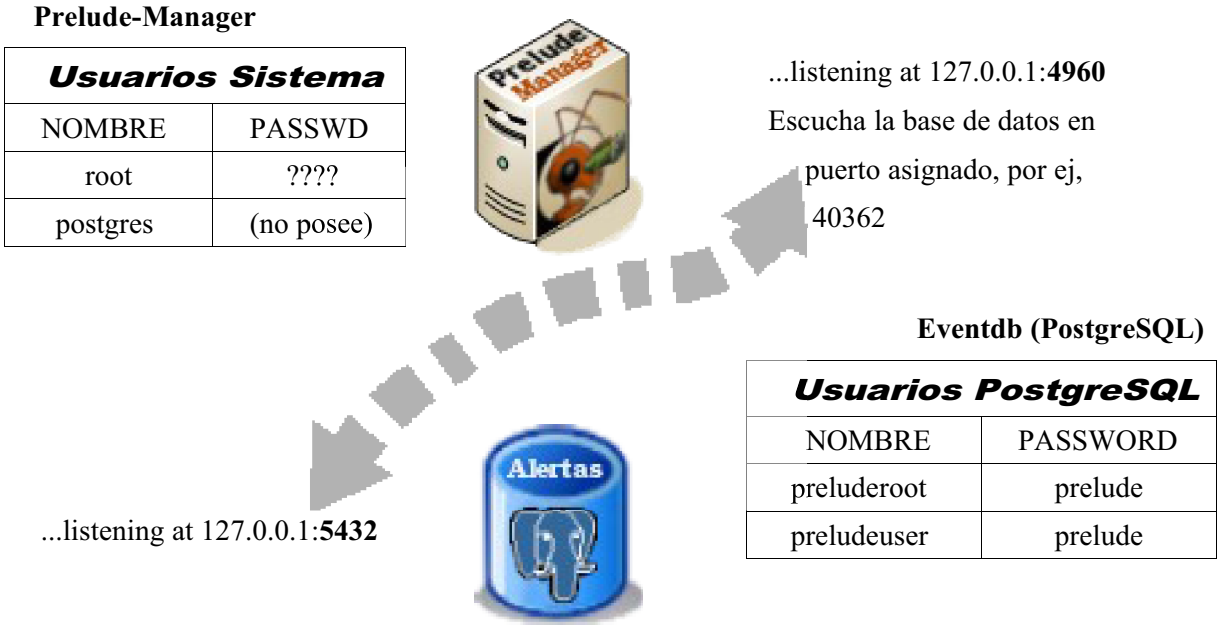

Figura 31.- Esquema de usuarios del sistema y de PostgreSQL.