Propuesta de un Mecanismo de Seguridad para el Intercambio de Datos de Usuario en Redes de Próxima Generación

Universidad del Cauca

Jaime Andrés Oliva Ortega
Fabio Joaquín Fuertes Montenegro

ANEXO A
ENDURECIMENTO DEL KERNEL DE LINUX EN DEBIAN

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, Octubre de 2009

TABLA DE CONTENIDO

LISTA DE FIGURAS

1. Introducción

Endurecer un software implica incrementar la protección, prevención y detección de amenazas de seguridad, en este caso propias de la ejecución de código malicioso sobre el Núcleo o Kernel del sistema operativo Linux, que pueden causar desde la obtención de privilegios administrativos sobre la máquina hasta el bloqueo de la misma. Para realizar esta tarea, se hará uso del parche Grsecurity el cual brinda las características de seguridad mencionadas.

Todos los pasos seguidos en este documento son ejecutados con privilegios de usuario *root* sobre el host *zealot,* el cual, cuenta con sistema operativo Debian 5 y Kernel 2.6.26-2 Instalado con la distribución. Cabe anotar que no es necesario volver a realizar todo el proceso para los otros equipos, ya que como resultado esta guía se obtendrá un paquete genérico de Debian, que puede instalarse fácilmente en otros equipos con este sistema operativo.

2. Requerimientos preliminares

Para compilar el Kernel, primero es necesario instalar algunos paquetes específicos como se muestra en la Figura 1.

Figura 1. Instalación de paquetes necesarios para la compilación del Kernel

Para realizar una configuración del Kernel en una consola grafica (*make menuconfig*), se instala el paquete mostrado en la Figura 2.

Figura 2. Instalación del paquete *libncourses5-dev*

Seguidamente, es necesario verificar si el paquete *initramfs-tool*, se encuentra instalado en el equipo (ver Figura 3).

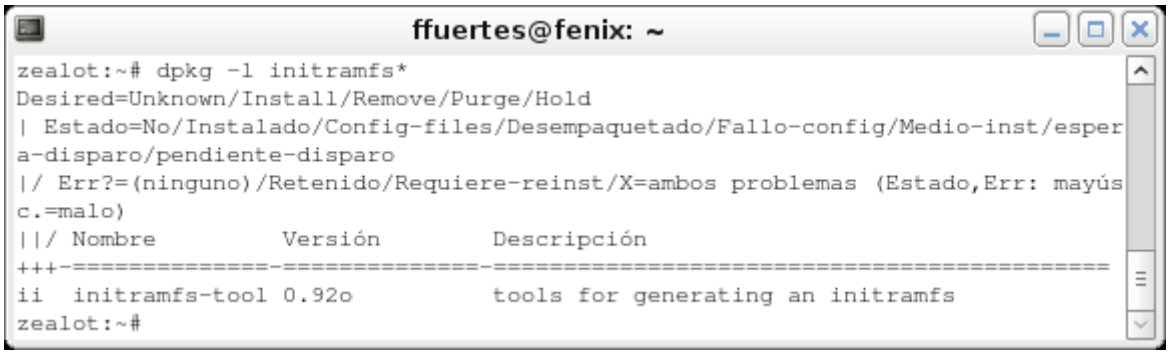

Figura 3. Verificación de la instalación del paquete initramfs-tool

Ahora se descargan las fuentes del Kernel 2.6.27, las cuales en el momento de llevar a cabo este procedimiento corresponden a la versión estable actualizada (ver Figura 4). Igualmente se descarga el parche Grsecurity para dicho Kernel (ver Figura 5).

Figura 4. Descarga de las fuentes del Kernel

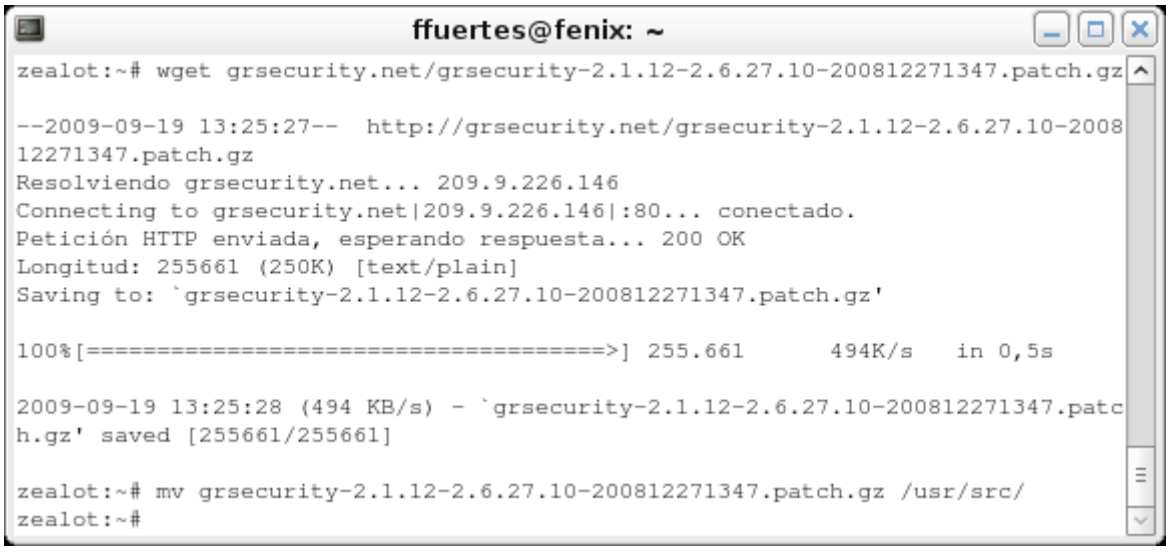

Figura 5. Descarga del parche Grsecurity

Los dos archivos descargados se mueven al directorio `/usr/src`. Posteriormente se procede a descomprimir el archivo del Kernel mediante el siguiente comando:

```
# tar xzvf linux-2.6.27.10.tar.gz
```

Se crea un enlace simbólico llamado `linux`, apuntando al directorio del nuevo Kernel para hacer un poco más sencillas las tareas posteriores mediante el siguiente comando:

```
# ln -s linux-2.6.27.10 linux
```

3. Parchar el Kernel

Ahora que todos los requerimientos preliminares se han cumplido se procede a parchar el Kernel siguiendo los siguientes pasos:

Mover el parche Grsecurity al nuevo directorio y entrar en este (ver Figura 6).

Figura 6. Mover el parche Grsecurity al directorio del Kernel

Descomprimir y parchar las fuentes del Kernel (ver Figura 7).

Figura 7. Descomprimir y parchar las fuentes del Kernel

Una vez se ha realizado este proceso, se han modificado las fuentes para realizar una instalación con elementos de seguridad avanzados.

4. Configurar el Kernel Endurecido

Para realizar la configuración, en este caso se utiliza el menú de consola *(make menuconfig),* utilizando el paquete *libncourses5-dev.* Sin embargo es posible realizar la configuración en una consola de solo texto (*make config*), o en interfaz gráfica (*make xconfig*).

Grsecurity tiene niveles de seguridad predefinidos: bajo, medio y alto, también puede ser configurado en un nivel personalizado donde se puede escoger habilitar o no, cada una de las opciones.

Para este caso se utilizará una configuración en nivel medio que asegura la protección contra las principales amenazas de ejecución de código malicioso existentes actualmente. Además, el nivel medio no es tan restrictivo como el alto, pues este puede deshabilitar algunos servicios necesarios en el prototipo. Los pasos necesarios para la configuración son los siguientes:

Lanzar el menú de consola (Figura 8):

Figura 8. Lanzar el menú de consola

Las figuras 9 a 11 muestran el proceso de configuración del nivel medio de Grsecurity en el menú de consola durante la puesta a punto de un Kernel a la medida.

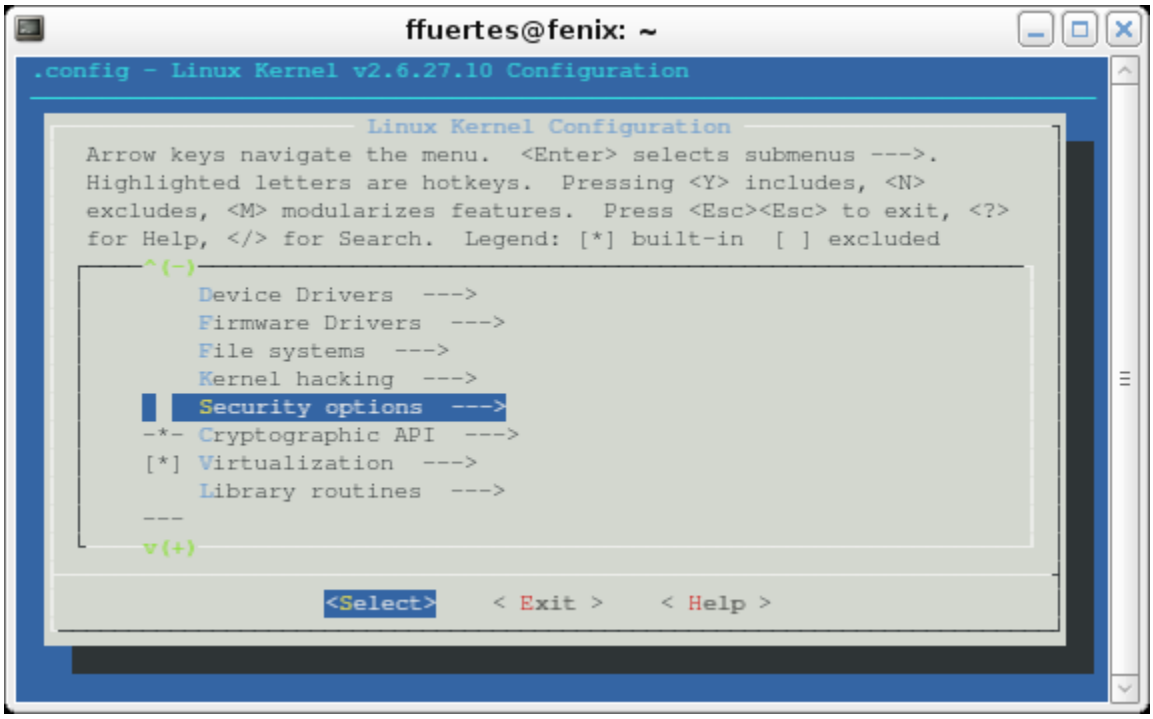

Figura 9. Escoger Security Options

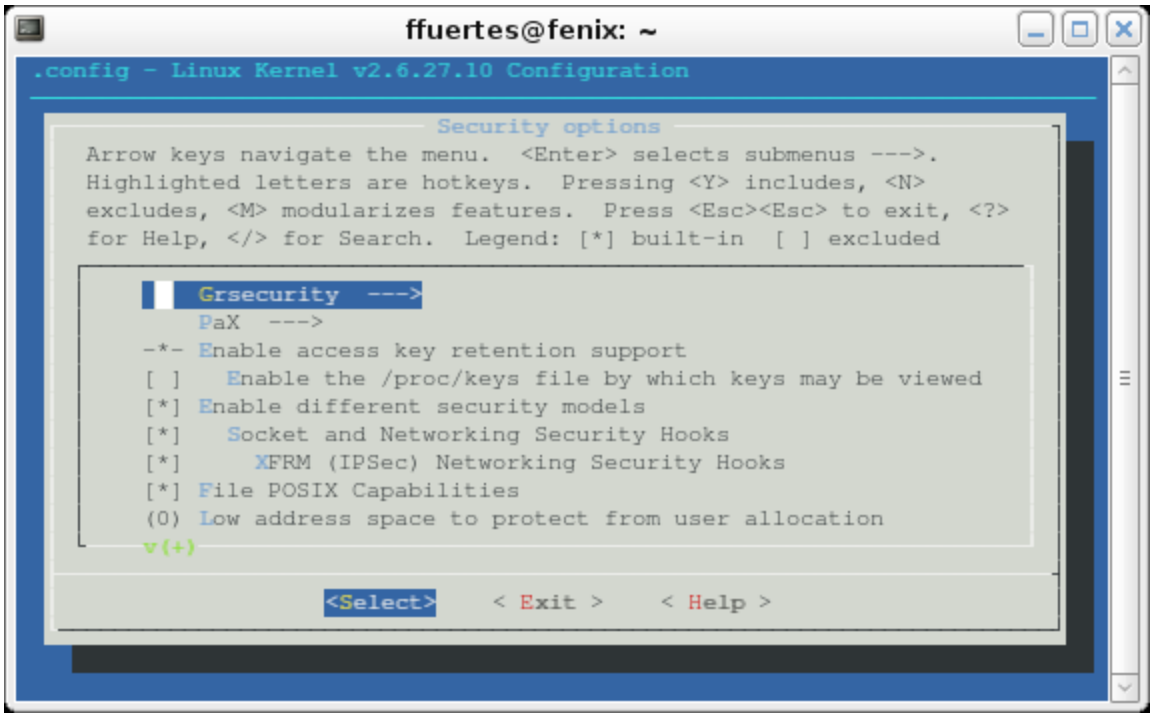

Figura 10. Escoger Grsecurity

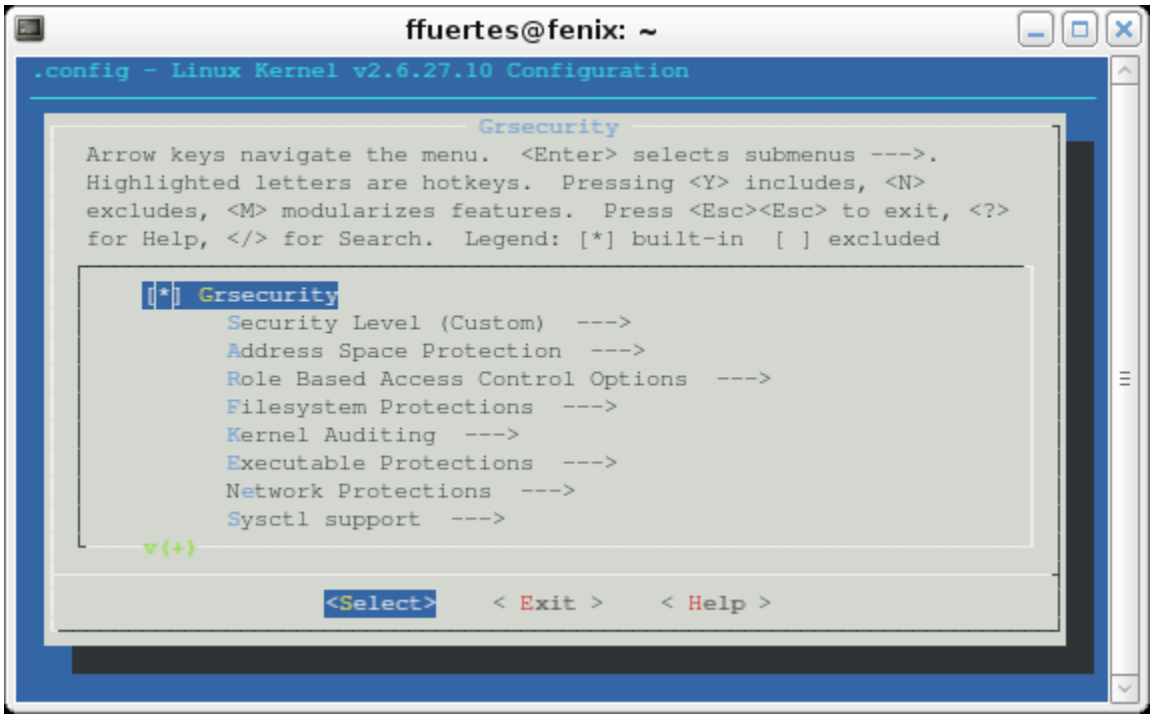

Figura 11. Marcar Grsecurity e ir a Security Level

Figura 12. Marcar el nivel Medium

5. Compilación del Kernel endurecido

Ahora se procede a compilar las fuentes ya configuradas con las opciones de seguridad de Grsecurity siguiendo los siguientes pasos:
Realizar limpieza de la configuración (Figura 13).

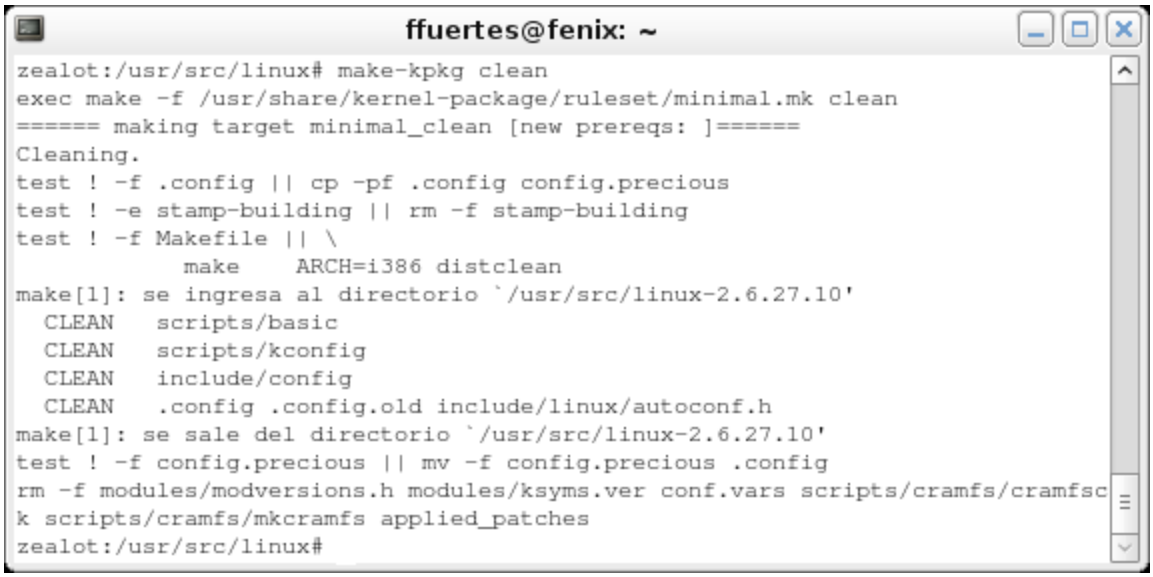

Figura 13. Realizar limpieza de la configuración previa a la compilación del Kernel

Lanzar la compilación como tal (Figura 14)

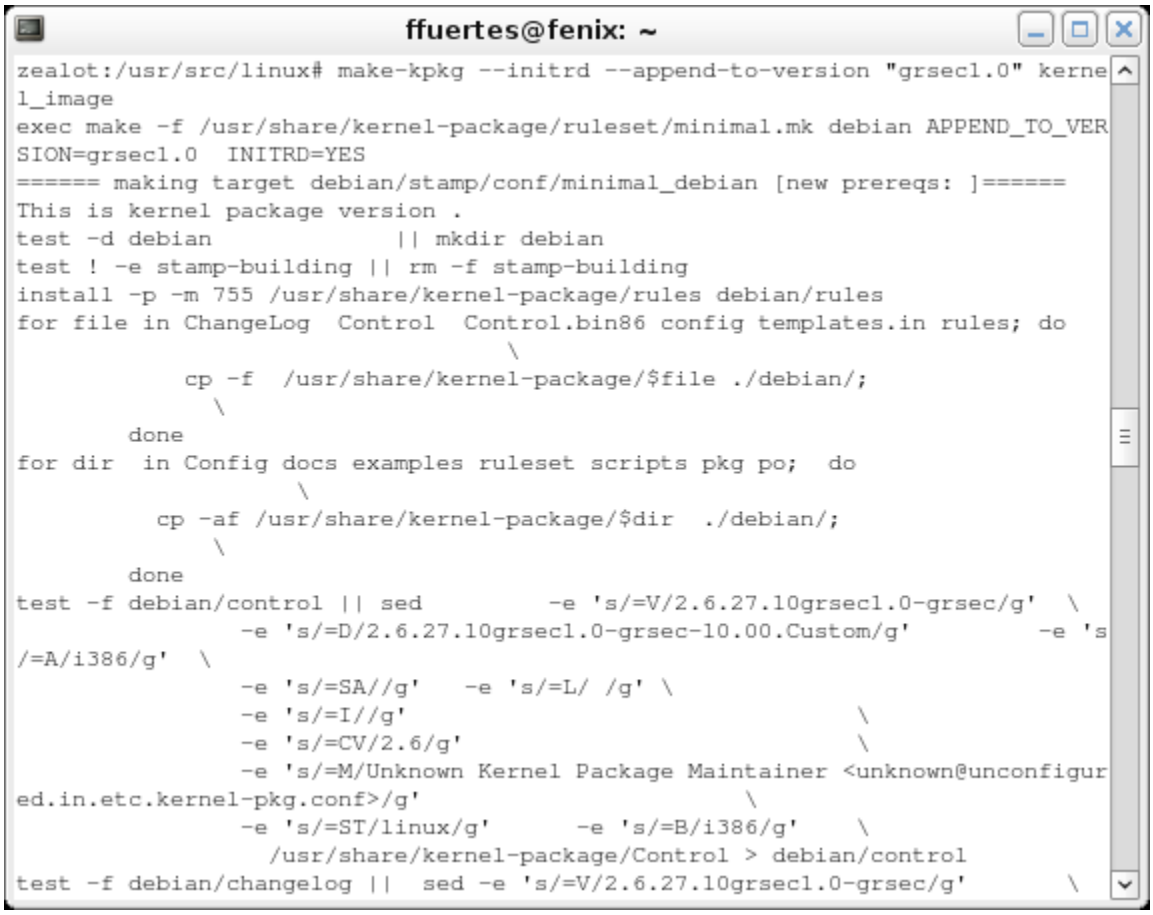

Figura 14. Compilación del Kernel endurecido

Esta compilación puede tardar varias horas, dependiendo de la velocidad de procesador y memoria RAM del equipo donde se esté realizando la operación. Para el caso del host zealot la duración de este proceso fue de alrededor de 3 horas, después de las cuales se termina el proceso como lo muestra la Figura 15.

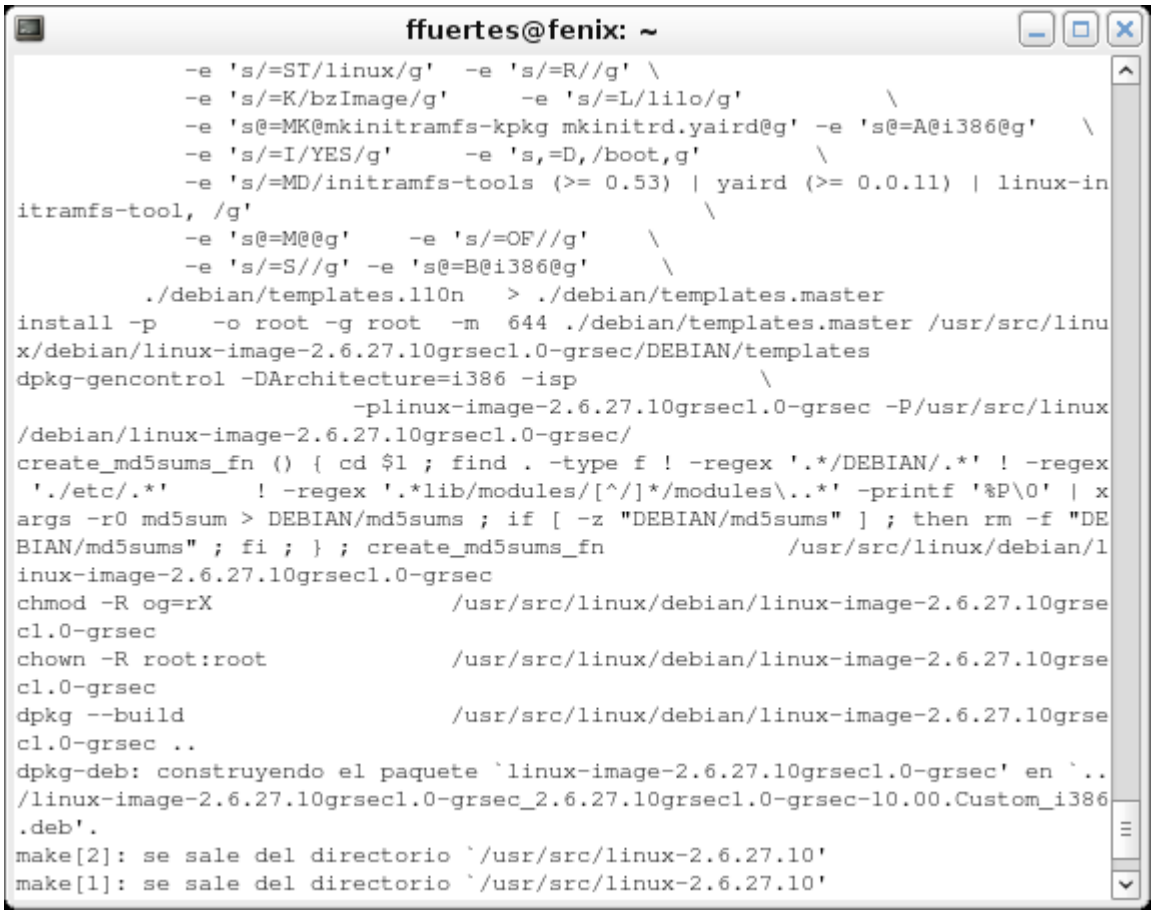

Figura 15. Terminación del proceso de compilación del Kernel endurecido

6. Instalación del Kernel Endurecido

El resultado de realizar el proceso de compilación es un paquete *.deb* (paquete instalable especifico para distribuciones de Linux basadas en Debian), que se encuentra en el directorio `/usr/src`, el cual puede instalarse del mismo modo que cualquier otro paquete para esta distribución mediante comando: `dpkg -i paquete.deb`, como lo muestra la Figura 16.

Figura 16. Instalación del Kernel endurecido

Durante la instalación se genera una imagen *initrd* la cual corresponde al Kernel instalado. Posteriormente se revisa las características de la imagen del Kernel generado (ver Figura 17).

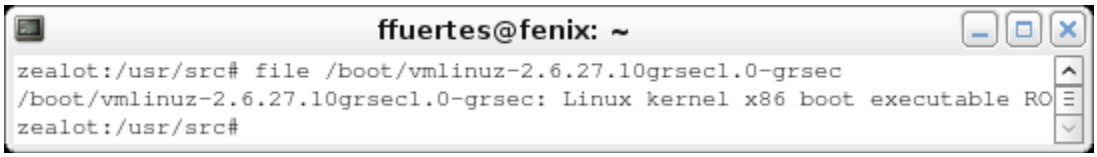

Figura 17. Verificación del archivo vmlinuz generado

Finalmente, es necesario reiniciar el equipo para que el nuevo Kernel entre en operación, ya que en el momento todavía se encuentra corriendo el anterior como lo muestra la Figura 18.

Figura 18. Verificación de la versión actual del Kernel antes del reinicio

Una vez el equipo reinicia se revisa la versión del Kernel mostrando que está corriendo la versión endurecida (ver Figura 19).

Figura 19. Verificación de la versión actual del Kernel después del reinicio

Como se mencionó al iniciar este documento, el hecho de que se obtiene un paquete instalable (*.deb*) después de la compilación, hace que no sea necesario volver a realizar todo el anterior proceso en cada host del prototipo, ya que simplemente se copia este archivo a los demás equipos y se instala como se mostró en el punto 5.
Las figuras 20 a 22 muestran la verificación de la instalación de este Kernel en los demás equipos del prototipo.

Figura 20. Verificación de la versión actual del Kernel en el host xeratul

Figura 21. Verificación de la versión actual del Kernel en el host dragoon

Figura 22. Verificación de la versión actual del Kernel en el host fenix