

# GRSECURITY - Seguridad en sistemas Linux mediante Detección, Prevención y Contención

Submitted by [freedom](#) on Wed, 25/05/2005 - 11:06. [Aplicaciones](#)

[Grsecurity](#) es una solución de seguridad a modo de parche de kernel que permite establecer múltiples comprobaciones, verificaciones y controles de una forma activa en nuestro sistema.

Estos mecanismos van desde la protección a nivel de funcionamiento del kernel, control de ejecución de las tareas en el stack, control de las actividades de los usuarios, permisos de ejecución en determinadas áreas del sistema, controles adicionales a la seguridad impuesta por chroot, y un largo etc.

A continuación vamos a comentar cada una de las opciones que permite esta solución, la traducción de los distintos apartados ha intentado ceñirse en lo posible a los textos originales. Se conservan los títulos originales de los apartados para facilitar su correcta identificación.

## GRSECURITY

Activando esta opción, nos permitirá habilitar el sistema Grsecurity, con lo cual dispondremos de un amplio abanico de posibilidades que podremos ir configurando para aumentar la seguridad del sistema. Es altamente recomendable activarla y consultar la ayuda para saber que opciones nos resultan interesantes.

- **Low additional security - Seguridad Adicional Baja**

Activando esta opción, activaremos varias de las opciones de grsecurity que permitirán alcanzar una elevada protección del sistema frente a un gran número de ataques, evitando que no exista ningún tipo de conflicto con el resto del sistema. Si se esta empleando algún software poco habitual, o si se encontrarán problemas de funcionamiento con niveles de seguridad más elevados, quizás esta sea la mejor opción.

Con esta opción, se habilitan las siguientes características de Grsecurity:

```
linking restrictions
fifo restrictions
random pids
enforcing nproc on execve()
restricted dmesg
random ip ids
enforced chdir("/") on chroot
```

- **Medium additional security - Seguridad Adicional Media**

Activando esta opción, varias características más de grsecurity además de las incluidas en el nivel bajo de seguridad serán habilitadas. Estas características proporcionan más seguridad al sistema, sin embargo en un porcentaje mínimo de

los casos pueden ser incompatibles con software muy antiguo o incorrectamente escrito. Si se va a activar esta opción, hay que asegurarse previamente de que el servicio auth (identd) esta ejecutándose con gid 10 (habitualmente el grupo wheel). con esta nivel de seguridad, se activarán las siguientes características, además de las proporcionadas por el nivel bajo de seguridad adicional:

```
random tcp source ports
failed fork logging
time change logging
signal logging
deny mounts in chroot
deny double chrooting
```

```
deny sysctl writes in chroot
deny mknod in chroot
deny access to abstract AF_UNIX sockets out of chroot
deny pivot_root in chroot
denied writes of /dev/kmem, /dev/mem, and /dev/port
/proc restrictions with special gid set to 10 (generalmente wheel)
address space layout randomization
removal of addresses from /proc//[maps|stat]
```

- **High additional security - Seguridad Adicional Alta**

Activando esta opción, muchas de las características de grsecurity se habilitarán, con lo cual el sistema quedará protegido frente a muchos tipos de ataques. El aumento de la seguridad incrementa también el número de incompatibilidades con software poco habitual que se encuentre en el sistema. Este nivel activa el sistema PaX, es recomendable leer [pax.grsecurity.net](http://pax.grsecurity.net) y documentarse sobre el proyecto PaX. Descarga chpax desde la web y ejecútalo en aquellos binarios que puedan suponer problemas de incompatibilidad con PaX. Además hay que recordar que con las restricciones de /proc activadas, has de ejecutar identd en el grupo wheel (gid 10). Este nivel de seguridad activa las siguientes protecciones además de las activadas en los niveles bajo y medio:

```
additional /proc restrictions
chmod restrictions in chroot
no signals, ptrace, or viewing processes outside of chroot
capability restrictions in chroot
```

```
deny fchdir out of chroot
priority restrictions in chroot
segmentation-based implementation of PaX
mprotect restrictions
kernel stack randomization
mount/unmount/remount logging
kernel symbol hiding
```

- **Customized additional security - Seguridad Adicional Personalizada**

Activando esta opción, se podrán configurar de forma individual todas las opciones de grsecurity, lo que permite activar muchas más características que no están cubiertas en los niveles de seguridad básicos. Estas características adicionales de grsecurity incluyen TPE, restricción de sockets, y el sistema sysctl. Es aconsejable leer la ayuda de todas las opciones para valorar si nos resultan útiles o no.

## PAX CONTROL

Las opciones que se presentan a continuación permiten especificar los métodos empleados para marcar los binarios del sistema para usar PaX.

- **Support soft mode - Modo de Soporte Soft**

Activando esta opción, ejecutaremos PaX en modo soft, que quiere decir, que las características de PaX no estarán activas por defecto, solo los ejecutables marcados de forma explícita. Tendremos que activar el soporte para PT\_PAX\_FLAGS ya que es el único modo de marcar los ejecutables para el uso del modo soft

El modo soft puede ser activado usando en la opción el siguiente código en la línea de comandos del kernel, pax\_softmode=1. Además se podrán controlar varias de las funcionalidades de PaX en tiempo de ejecución a través de las entradas en /proc/sys/kernel/pax.

- **Use ELF program header marking - Utilizar el marcado en las cabeceras ELF**

Activando esta opción se podrán controlar las características de PaX para cada ejecutable básico a través de la utilidad paxctl disponible en [pax.grsecurity.net](http://pax.grsecurity.net). Los elementos de control serán leídos de la cabecera ELF específica del programa (PT\_PAX\_FLAGS). Este marcado tiene las ventajas de soportar ambos métodos soft y estar plenamente integrado dentro de la solución ( el parche para binutils esta disponible en [pax.grsecurity.net](http://pax.grsecurity.net)).

Hay que considerar que si además se activa el soporte legacy EI\_PAX marking este será sobrescrito por el marcado PT\_PAX\_FLAGS.

- **MAC system integration - Integración del sistema MAC**

Las listas de control de acceso obligatorio tiene la opción de controlar los parámetros de PaX para cada ejecutable base, selecciona el método soportado para el sistema.

- none - ninguno: Si el sistema MAC no interactua con Pax.
- direct - directo : Si el sistema MAC define pax\_set\_flags() el mismo.
- hook - gancho: Si el sistema MAC usa la llamada apax\_set\_flags\_func.

NOTA: Esta opción es únicamente para desarrolladores / integradores.

## **ADDRESS SPACE PROTECTION - PROTECCIÓN DEL ESPACIO DE DIRECCIONES**

Las opciones que se detallan a continuación permiten establecer la configuración de PaX, además de un reducido número de otras funcionalidades. PaX es un proyecto separado de Grsecurity pero es una parte crucial de la filosofía de grsecurity. PaX previene el abuso de los desbordamientos de búfer, además de dotar de aleatoriedad a la gestión de procesos en memoria, ambas medidas son increíblemente efectivas en la protección contra atacantes.

- **Enforce non-executable pages - Impedir la ejecución en páginas**

Por diseño algunas arquitecturas no permiten la protección de páginas contra ejecución o incluso si lo hacen, Linux no utiliza esta característica. En la práctica significa que si una página es de lectura (como la pila o la cola) también es ejecutable.

Existe una conocida técnica de exploits que utiliza este hecho junto con errores frecuentes de programación donde es posible introducir código arbitrario en algún lugar de la memoria del programa en ejecución (habitualmente la pila o la cola) y después ejecutarlo.

Si el programa vulnerable está ejecutándose con diferentes privilegios (habitualmente superiores) a los del intruso, entonces es posible que el intruso obtenga un incremento en sus privilegios, lo que se conoce como ataque de escalada de privilegios. (Algunos de estos casos son obtener una cuenta de root, escribir en ficheros a los cuales no debería de tener acceso, etc.)

Activando esta opción, podremos escoger entre distintas características que previenen la inyección y ejecución de código ajeno al programa.

Esta opción provocará que algunos programas que confían en el antiguo comportamiento y esperan que la memoria reservada dinámicamente a través de la familia de funciones `malloc()` sea ejecutable (que no lo es). Algunos ejemplos destacados son el servidor XFree86 4.x server, el entorno de ejecución de java o wine.

- **Paging based non-executable pages - Paginación basada en páginas no ejecutables**

Esta implementación está basada en la característica de paginación de la CPU. En i386 tiene un impacto variable en el rendimiento, en función del patrón de uso de memoria. Es aconsejable probar las aplicaciones antes de usar esta funcionalidad en entornos de producción. En alpha, parisc, sparc y sparc64 no hay impacto en el rendimiento. En ppc hay un leve impacto en el rendimiento.

- **Segmentation based non-executable pages - Segmentación basada en páginas no ejecutables**

Esta implementación esta basada en la característica de segmentación de la CPU y tiene poco impacto en el rendimiento, sin embargo las aplicaciones serán limitadas a 1.5 GB de espacio de direcciones en lugar de 3GB normales.

- **Emulate trampolines - Emulación de trampolines**

Existen algunos programas y librerías que por una u otra razón, intentan ejecutar pequeños trozos de código de las paginas de memoria no ejecutables. Los ejemplos más significativos son el código de retorno del gestor de señales, generado por el propio kernel y los trampolines del GCC.

Si habilitamos las opciones `CONFIG_GRKERNSEC_PAX_PAGEEXEC` o `CONFIG_GRKERNSEC_PAX_SEGMEXEC` entonces estos programas dejarán de funcionar con esta configuración de kernel.

Como solución, se puede activar esta opción, y usar las utilidades `chpax` o `paxctl` para realizar la emulación de trampolines para los programas a los que afecte esta restricción manteniendo la protección de páginas no ejecutables.

En arquitecturas `parisc` y `ppc` se TIENE que habilitar esta opción además de `EMUSIGRT`, si no el sistema no arrancará.

Alternativamente se puede decir que No (desactivarla) en esta opción y emplear las utilidades `chpax` o `paxctl` para desactivar las opciones de `CONFIG_GRKERNSEC_PAX_PAGEEXEC` y `CONFIG_GRKERNSEC_PAX_SEGMEXEC` en los ficheros afectados.

NOTA: activando esta opción "puede" crearse una fisura que podría ser utilizada para intentar vulnerar la seguridad del sistema. Por lo tanto, la mejor solución es la de no tener en el sistema ficheros que requieran esta opción. Esto puede lograrse utilizando `libc5` ( que confía en el código de retorno del gestor de señales ) y no utilizar o reescribir programas que hacen uso de la implementación de función por jerarquías de GCC. Los usuarios expertos pueden arreglar directamente el propio GCC para implementar las llamadas a las funciones jerárquicas de forma que no interfieran con PaX.

- **Automatically emulate sigreturn trampolines - Emulación Automática de señales de retorno en trampolines**

Activando esta opción, el kernel estará detectando y emulando las señales de retorno de trampolines que se estén ejecutando en la pila y que de otra forma provocarían la finalización del proceso.

Esta opción esta pensada como solución temporal para usuarios con versiones compatibles de `libc` (`libc5`, `glibc 2.0`, `uClibc` previo a 0.9.17, entorno de ejecución de `Modula-3`, etc) o ejecutables enlazados a la misma, básicamente todo aquello que no especifica su propia función `SA_RESTORER` en la memoria ejecutable normal como hace `glibc 2.1+`

En parisc y ppc SE TIENE que activar esta opción, de otra forma el sistema no se iniciará.

NOTA: esta característica no puede ser desactivada por ejecutable base y como abre una pequeña brecha en la protección establecida por las paginas no ejecutables, la mejor solución es no tener ficheros en el sistema que requieran tener activa esta función.

- **Restrict mprotect() - Restricción de mprotect()**

Activando esta opción se evitará que los programas puedan realizar:

- Modificación del estado de ejecución de las páginas de memoria que originalmente no estaban creadas como ejecutables.
- Modificación del estado de las páginas marcadas como de solo lectura a escritura.
- Creación de páginas ejecutables desde memoria anónima.

Se debería de activar esta opción para completar la protección facilitada por el cumplimiento de páginas no ejecutables, `enforcement of non-executable pages`

NOTA: Se puede utilizar la utilidad `chpax` para controlar esta característica en un fichero base. `chpax` esta disponible en [pax.grsecurity.net](http://pax.grsecurity.net)

- **Disallow ELF text relocations - Desactivar la relocalización de texto en ELF**

Las restricciones relativas a páginas no ejecutables y `mprotect()` son efectivas en la prevención de la introducción de nuevo código ejecutable en el espacio de direcciones de una tarea. Aunque siguen existiendo dos caminos para realizar este tipo de ataque: ejecutar código ya existente en la tarea con lo que se puede o crearlo y realizar `mmap()` sobre un fichero que contenga dicho código o haber realizado `mmap()` sobre una librería ELF existente que no tenga código independiente de la petición en ella y use `mprotect()` para que sea posible escribir en ella y copiar ahí el código.

Mientras que la protección frente al anterior método está lejos de la protección que ofrece PaX, el último método puede prevenirse teniendo librerías únicamente **ELF PIC** en el sistema (que no tengan que recolocar su código). Si este es el caso, entonces se recomienda activar esta opción, si no es el caso, hay que tener mucho cuidado ya que el sistema podría no iniciarse (por ejemplo, algunos módulos de PAM están compilados incorrectamente como non-PIC por defecto).

NOTA: Si se están usando ejecutables ELF dinámicos (como se sugiere cuando se usa ASLR) entonces hay que asegurarse que los ficheros han sido enlazados usando la versión PIC de `crt1` ( el paquete `et_dyn.zip` al que se hace referencia ha sido actualizado para dar soporte a estas funcionalidades).

- **Enforce non-executable pages - Establecer las páginas como no ejecutables**

Esto es la parte de kernel land equivalente a PAGEEXEC y MPROTECT, es decir, activando esta opción será más complicado insertar código externo en la propia memoria del kernel.

- **Address Space Layout Randomization - Aleatoriedad en la disposición del espacio de direcciones**

Algunas si no la mayoría de las técnicas de exploit se apoyan en el conocimiento de ciertas direcciones de los programas a los que atacan. Las siguientes opciones permitirán al kernel aplicar cierto grado de aleatoriedad a partes específicas de los programas obligando por lo tanto a forzar una averiguación de las direcciones en la mayoría de los casos.

Cualquier intento de averiguación de estas direcciones que sea fallido, provocará seguramente la parada de la tarea lo que permitirá al kernel detectar estos intentos y reaccionar ante ellos. El propio PaX en si mismo no incorpora mecanismos que reaccionen ante este tipo de actividad, se recomienda por lo tanto utilizar las funcionalidades de Grsecurity para la detección de fallos o incluso desarrollar uno propio.

Activando estas opciones, se podrá seleccionar la aleatoriedad de las siguientes áreas:

- la cima de la pila de tareas de kernel
- la cima de la pila de tareas en userland
- la dirección base para las peticiones de `mmap()` que no especifiquen una (incluyendo todas las librerías)
- la dirección base del ejecutable principal

Es muy recomendable activar esta opción ya que la aleatoriedad en la disposición del espacio de direcciones no tiene impacto sobre el rendimiento y proporciona una protección muy efectiva.

NOTA: Se puede usar `chpax` o `paxctl` para controlar la mayoría de estas funcionalidades en cada fichero.

- **Randomize kernel stack base - Selección Aleatoria de la base de la pila del kernel**

Activando esta opción, el kernel podrá generar de forma aleatoria en cada llamada al sistema la pila del kernel para cada tarea. Esta opción forzará a averiguar dicha dirección si se quiere hacer mal uso de la misma, y además previene que se haga un uso inadecuado de la información que podría estar contenida en esta dirección.

Desde que la pila del kernel es un recurso algo escaso, la generación aleatoria puede generar desbordamientos de pila no esperados, por lo tanto es necesario probar el sistema cuidadosamente. Hay que tener presente que una vez activada esta opción en la configuración del kernel, no puede ser desactivada en ficheros concretos.

- **Randomize user stack base - Generación aleatoria de direcciones base en la pila de usuario**

Activando esta opción el kernel generará de forma aleatoria cada una de las tareas de la pila en userland. La generación aleatoria se hará en dos pasos donde el segundo aplicará una gran cantidad de modificaciones en el top de la pila y causará problemas a aquellos programas que usen gran cantidad de memoria ( más de 2.5 Gb si **SEGMEXEC** no esta activo, o 1.25 GB si lo esta). Por esta razón el segundo paso puede ser controlado mediante **chpax** o **paxctl** en un archivo base.

- **Randomize ET\_EXEC base - Generación aleatoria de la dirección base en ET\_EXEC**

Activando esta opción, el kernel también generará de forma aleatoria la dirección base de los ejecutables ELF ET\_EXEC. Esto se realiza ubicando el ejecutable en memoria de una forma especial que también permite detectar los intentos de ejecución de código no autorizado. Como esta ubicación especial causa degradación del rendimiento y la detección de ataques además puede provocar falsos positivos, es importante probar los ejecutables cuidadosamente cuando esta funcionalidad esta activada.

Esta opción esta pensada como solución temporal hasta que se reenlacen los programas como ficheros ELF dinámicos.

NOTA: Para controlar esta funcionalidad por fichero se puede utilizar **chpax** o **paxctl**.

- **Allow ELF ET\_EXEC text relocations - Permitir la relocalización de texto en ELF ET\_EXEC**

En algunas arquitecturas como alpha hay aplicaciones incorrectamente creadas que requieren relocalizaciones de texto y que no podrán funcionar sin activar esta opción. Si eres un usuario de alpha, deberías de activar esta opción y deshabilitarla una vez te hayas asegurado de que ninguna de tus aplicaciones la necesita.

- **Automatically emulate ELF PLT - Emulación Automática de ELF PLT**

Activando esta opción el kernel detectará y emulará de forma automática las entradas de **Procedure Linkage Table** en los ficheros ELF. En algunas arquitecturas dichas entradas están en memoria donde se puede escribir, y convertirlas en no ejecutables puede provocar la finalización de la tarea. Por lo tanto es obligatorio activar esta opción en alpha, parisc, ppc, sparc y sparc64, de otra forma el sistema podría no iniciarse.

NOTA: Esta funcionalidad abre una brecha en sistema de protección aportado por la paginación no ejecutable, por lo tanto la solución más correcta pasa por

modificar la aplicación de forma que no genere un PLT en el que pueda escribirse.

- **Randomize mmap() base - Generación aleatoria de la dirección base de mmap()**

Activando esta opción, el kernel utilizará una dirección de memoria base generada de forma aleatoria en las peticiones de `mmap()` que no vengan ya especificadas. Como resultado todas las librerías cargadas de forma dinámica aparecerán en direcciones aleatorias y por lo tanto serán más difícil emplear técnicas en las que se intenta ejecutar código de una librería para otros propósitos. (P.ej: ejecutar una shell desde un programa vulnerable que se este ejecutando con privilegios elevados).

Además, si un programa es reenlazado como un fichero ELF dinámico, su dirección base también será generada de forma aleatoria, completando la aleatoriedad del espacio de direcciones disponible. Intentar atacar estos programas se convierte en un juego de adivinación. Se puede encontrar un ejemplo de esto en [et\\_dyn.zip](#) y ejemplos prácticos en [grsec-gcc-specs.tar.gz](#)

NOTA: para controlar esta característica en ficheros individuales se pueden emplear las herramientas `chpax` o `paxctl`.

- **Deny writing to /dev/kmem, /dev/mem, and /dev/port - Prohibición de escritura a /dev/kmem, /dev/mem y /dev/port**

Activando esta opción, se impedirá la escritura en `/dev/kmem` y `/dev/mem` a través de `mmap` u otra forma de modificar el kernel en ejecución. Tampoco se permitirá la apertura de `/dev/port/`. Si el soporte para módulos se encuentra desactivado, activando esta opción se cerraran cuatro maneras que se usan para insertar código en el kernel en ejecución. Incluso con estas restricciones activas, se recomienda encarecidamente el uso del sistema **RBAC**, ya que sigue siendo posible modificar el kernel a través de acceso privilegiado de I/O a través de `ioperm/iopl`. Si no se esta usando **XFree86**, este último riesgo se evitará activando la opción `Disable privileged I/O`. Aunque ninguna aplicación escribe de forma legitima en `/dev/kmem`, **XFree86** necesita escribir en `/dev/mem`, pero únicamente a la memoria de video, que es lo único que se permite con esta opción. Si `/dev/kmem` y `/dev/mem` estan marcados sin `PROT_WRITE`, no podrán usar `mprotect` con `PROT_WRITE` despues.

Se recomienda encarecidamente activar esta opción si se cumplen los requisitos anteriores.

- **Disable privileged I/O - Desabilitar I/O privilegiada**

Activando esta opción, todas las llamadas `ioperm` y `iopl` devolverán un error. `ioperm` y `iopl` pueden ser usadas para modificar el kernel en ejecución. Desafortunadamente, algunos programas utilizan estas llamadas para funcionar correctamente, las más

destacadas

son **XFree86** y **hwclock**. **hwclock** puede seguir funcionando teniendo soporte RTC en el kernel, por lo que hay que activar `CONFIG_RTC` si queremos usar esta opción y asegurarnos de que funcione. **XFree86** tampoco funcionará aún teniendo activada esta opción, por lo tanto se recomienda que **NO SE ACTIVE** esta opción si se usa **XFree86**. Si se usa **XFree86** y se quiere establecer una protección contra la modificación del kernel, la recomendación pasa por emplear el sistema **RBAC**.

- **Hide kernel symbols - Ocultar símbolos del kernel**

Activando esta opción, la obtención de información de los módulos cargados y la información de los símbolos de sistema a través de `syscall` estarán restringidas a usuarios con el módulo `CAP_SYS_MODULE`. Esta opción solo será efectiva cuando se cumplan las siguientes condiciones:

- 1) El kernel usando `grsecurity` no está precompilado por alguna distribución.
- 2) Se está usando el sistema **RBAC** y ocultando otros ficheros como la imagen del kernel y el `System.map`
- 3) Están activas las restricciones adicionales de `/proc`, que eliminan el `/proc/kcore`.

Si se cumplen las restricciones anteriores, esta opción permitirá establecer una protección útil contra el abuso de `overflows` de kernel tanto remotos como locales y contra vulnerabilidades de lectura/escritura arbitraria.

## **FILESYSTEM PROTECTIONS - PROTECCIONES DEL SISTEMA DE FICHEROS**

En este apartado se configuran las características relacionadas con el sistema de ficheros, incluyendo restricciones en el `/proc`, que hacen que un usuario pueda ver únicamente sus procesos, protecciones de ejecución en `/tmp`, además de cambiar las restricciones las rutas, lo que incrementa la seguridad como es el caso de (`chroot`) en las aplicaciones.

- **Proc Restrictions - Restricciones de proc**

Activando esta opción, los permisos del sistema de ficheros `/proc` serán modificados para aumentar la seguridad del sistema y la confidencialidad. Dependiendo de las opciones que se activen, se podrá permitir que únicamente los usuarios puedan ver sus propios procesos, o escoger un grupo que pueda ver todos los procesos y ficheros normalmente restringidos para `root` seleccionando `restrict to user only`.

NOTA: Si se está ejecutando `identd` con un usuario no `root`, se tendrá que ejecutar con el grupo que se especifique aquí.

- **Restrict /proc to user only - Restricción de /proc solo a usuarios**

Activando esta opción, los usuarios que no sean root, podrán ver únicamente sus procesos, lo que evitará que puedan acceder a información de red, enlaces de kernel e información de los módulos.

- **Restrict /proc to user and group - Restricción de acceso a /proc a usuarios y grupos**

Activando esta opción, se podrá seleccionar un grupo para ver los procesos, y la información de red, kernel y símbolos.

Esta opción resulta útil para ejecutar `identd` con un usuario distinto de root.

- **Remove addresses from /proc/pid/[maps|stat] - Eliminar direcciones de /proc/pid/[maps|stat]**

Activando esta opción, los ficheros `/proc//maps` y `proc//stat` no facilitarán información sobre las direcciones o sus

correspondencias si las características de PaX que confían en las direcciones aleatorias están habilitadas en la tarea.

Si se usa PaX es muy recomendable que actives esta opción, ya que cierra un fallo de seguridad que hace que el sistema de ASLR sea ineficaz con binarios `suid`.

- **Additional proc restrictions - Restricciones adicionales en proc**

Activando esta opción, se aplicarán restricciones adicionales en `/proc` que evitarán que los usuarios no privilegiados puedan ver información sobre la `cpu` y los dispositivos.

- **Linking restrictions - Restricción de enlaces**

Activando esta opción, se evitarán los exploits que utilizan el `/tmp`, ya que los usuarios no podrán seguir enlaces simbólicos que sean propiedad de otros usuarios en directorios de lectura para todos el mundo `+t` (por ejemplo, el propio `/tmp`), a menos que el propietario del enlace simbólico sea el propietario del directorio. Los usuarios tampoco podrán crear enlaces duros a ficheros de los que no sean propietarios.

Si la opción de `sysctl` está activa, se creará dentro del `sysctl` una opción con el nombre `linking_restrictions`

- **FIFO restrictions - Restricciones FIFO**

Activando esta opción, los usuarios no podrán escribir en los FIFOs que no posean en directorios de lectura para todo el mundo `+t` (por ejemplo, `/tmp`), a menos que el propietario del FIFO sea el mismo que el propietario del directorio donde está el FIFO.

Si la opción de `sysctl` está activada, se creará dentro del `sysctl` una entrada con el nombre `fifo_restrictions`

- **Chroot jail restrictions - Restricciones en jaulas Chroot**

Activando esta opción, Grsecurity nos permitirá seleccionar distintas opciones que harán que la ruptura de entornos chroot sea mucho más complicada. Si no existen incompatibilidades de software con las siguientes opciones, se recomienda habilitar todas las siguientes.

- **Deny access to abstract AF\_UNIX sockets out of chroot - Prohibición de acceso a sockets AF\_UNIX fuera de chroot**

Activando esta opción, los procesos que se estén ejecutando dentro de **chroot** no serán capaces de conectar a **abstract Unix Domain Sockets** (que no pertenezcan al sistema de ficheros) enlazados fuera del chroot.

Es recomendable activar esta opción.

Si la opción `sysctl` esta habilitada, se creará dentro de `sysctl` una opción con el nombre `chroot_deny_unix`.

- **Deny shmat() out of chroot - Prohibición de shmat() fuera de chroot**

Activando esta opción, los procesos que se estén ejecutando dentro de chroot, no podrán adjuntarse a segmentos compartidos de memoria que hayan sido creados fuera de la jaula chroot.

Es recomendable activar esta opción.

Si la opción `sysctl` esta activada, se creará dentro de `sysctl` una opción con el nombre `chroot_deny_shmat`.

- **Protect outside processes - Protección de procesos externos a chroot**

Activando esta opción, los procesos dentro de chroot no podrán matar, enviar señales con **fcntl**, **ptrace**, **capget**, **setpgid**, **getpgid**, **getsid**, o ver algún proceso fuera del chroot.

Si la opción `sysctl` esta activada, se creará dentro de `sysctl` una opción con el nombre `chroot_findtask`.

- **Deny mounts in chroot - Prohibición de uso de mount en chroot**

Activando esta opción, los procesos que se encuentren dentro de la jaula chroot no podrán montar o remontar sistemas de ficheros.

Si la opción `sysctl` esta activada, se creará dentro de `sysctl` una opción con el nombre `chroot_deny_mount`

- **Deny pivot\_root in chroot - Prohibición de pivot\_root en chroot**

Activando esta opción, los procesos dentro del chroot no podrán usar una función llamada `pivot_root` incluida en Linux 2.3.41. El funcionamiento es similar a chroot en el sentido de que modifica el sistema de ficheros de root.

Esta función puede ser utilizada en entornos chroot para intentar romper el chroot, y por lo tanto no debería estar permitida.

Si la opción `sysctl` está activada, se creará dentro de `sysctl` una opción con el nombre `chroot_deny_pivot`.

- **Deny double-chroots - Prohibición de doble-chroot**

Activando esta opción, los procesos que se encuentren dentro de la jaula chroot, no podrán volver a crear un chroot fuera del primer chroot. Este es un método empleado para romper los sistemas de jaula chroot.

Si la opción de `sysctl` esta activada, se creará dentro del `sysctl` una opción con el nombre `chroot_deny_chroot`

- **Deny fchdir outside of chroot - Prohibir fchdir fuera de chroot**

Activando esta opción, se evitarán las rupturas de chroot a través de los métodos basados en ejecutar `fchdir` a un descriptor de un fichero del proceso chroot que apunta a un directorio fuera del sistema de ficheros.

Si la opción `sysctl` esta activada, se creará dentro de `sysctl` una opción con el nombre `chroot_deny_fchdir`.

- **Enforce chdir("/") on all chroots - Establecer chdir("/") en todos los chroots**

Activando esta opción, el directorio de trabajo actual de todas las aplicaciones que sean creadas en chroot serán fijados al directorio raíz del chroot.

La página 2 del manual de chroot dice:

Esta llamada no modifica el directorio de trabajo actual, de modo que `'.'` puede estar fuera del directorio raíz en `'/'`. En concreto, el superusuario puede salir de una jaula chroot ejecutando `mkdir foo;chroot foo; cd ...`

Es recomendable activar esta opción, puesto que se sabe que no afecta al funcionamiento de ninguna aplicación.

Si la opción `sysctl` esta habilitada, se creará dentro de `sysctl` una opción con el nombre `chroot_enforce_chdir`.

- **Deny (f)chmod +s in chroot - Prohibir (f)chmod +s en chroot**

Activando esta opción, los procesos que se encuentren dentro de chroot no podrán ejecutar `chmod` o `fchmod` ficheros para hacerles tener bits de **suid** o **sgid**. Este mecanismo establece otra protección contra otros métodos de ruptura de chroot.

Si la opción `sysctl` esta habilitada, se creará dentro de `sysctl` una opción con el nombre `chroot_deny_chmod`.

- **Deny mknod in chroot - Prohibir mknod en chroot**

Activando esta opción, los procesos dentro del chroot no podrán ejecutar `mknod`. El problema de emplear `mknod` dentro de chroot es que puede permitir crear una entrada de dispositivo que tenga la misma ruta física que uno del sistema, pudiendo ser esta cualquier cosa, desde el dispositivo de consola a un dispositivo para el disco duro (que se podrá usar para obtener información o borrar el disco).

Se recomienda tener activada esta opción, a no ser que provoque incompatibilidades con alguna aplicación.

Si la opción `sysctl` esta habilitada, se creará dentro de `sysctl` una opción con el nombre `chroot_deny_mknod`.

- **Restrict priority changes in chroot - Restricción de cambio de prioridades en chroot**

Activando esta opción, los procesos que se encuentren dentro del chroot no podrán aumentar su prioridad en el chroot, o modificar la prioridad de procesos fuera de la jaula chroot. Esta opción aporta mayor seguridad que si elimináramos `CAP_SYS_NICE` del conjunto de capacidades del proceso.

Si la opción de `sysctl` esta activada, se creará una opción dentro de `sysctl` con el nombre `chroot_restrict_nice`

## **KERNEL AUDITING - REGISTRO DE ACTIVIDAD EN EL KERNEL**

En esta sección se pueden configurar varias opciones de auditoria. Es decir, aquellas funcionalidades que aunque no aportan seguridad en si mismas, representan información útil para el administrador que puede tener relevancia en materia de seguridad.

- **/proc//ipaddr support - Soporte para /proc//ipaddr**

Activando esta opción, se añadirá una nueva entrada a cada directorio `/proc/` que contendrá la dirección IP de la persona que esta utilizando la tarea. Esta dirección IP se trasmite a través de los sockets TCP locales y de los `AF_UNIX`.

Esta información puede ser útil para que los IDS/IPses puedan realizar una respuesta remota a un ataque local. Esta entrada puede ser leída únicamente por el propietario del proceso ( y el usuario root si tiene activado `CAP_DAC_OVERRIDE`, que puede ser desactivado a través del sistema RBAC), y que no creará así problemas de privacidad.

- **Single group for auditing - Registro para un sólo Grupo**

Activando esta opción, las características de registro de `exec`, `chdir`, `(un)mount` e `ipc` sólo funcionarán en un grupo específico. Esta opción es recomendable únicamente si se quiere vigilar un cierto grupo de usuarios en lugar de tener una enorme cantidad de logs de todo el sistema.

Si la opción `sysctl` esta habilita, se creará una opción dentro de `sysctl` con el nombre `audit_group`.

- **GID for auditing - GID para registro**

En esta opción se especificará el GID que será objetivo de la auditoria de kernel. Recuerda añadir los usuarios que se quieren vigilar al grupo especificado aquí.

Si la opción `sysctl` esta activada, no importará lo que se seleccione aqui. Tendrás que especificar el GID en el script de arranque enviando con `echo` el GID a la correspondiente entrada del `/proc`. Consulta la ayuda de las opciones de `sysctl` para más información.

Si la opción `sysctl` esta activada, se creará una entrada dentro de `sysctl` con el nombre `audit_gid`.

- **Chdir logging - Registro de Chdir**

Activando esta opción, todas las llamadas a `chdir()` serán registradas.

Si el sistema `sysctl` esta activado, se creará una opción dentro de `sysctl` con el nombre `audit_chdir`.

- **(Un)Mount logging - Registro de (un)mount**

Activando esta opción, todos los comandos `mount` y `unmount` serán registrados.

Si la opción `sysctl` esta activa, se creará una entrada dentro de `sysctl` con el nombre `audit_mount`.

- **IPC logging - Registro de IPC**

Activando esta opción, la creación y eliminación de colas de mensajes, semáforos, y memoria compartida será registrada.

Si la opción de `sysctl` esta activada, se creará dentro de `sysctl` una opción con el nombre `audit_ipc`.

- **Exec logging - Registro de Exec**

Activando esta opción, todas las llamadas a la función `execve()` quedarán registradas, (todas las ejecuciones serán registradas, ya que el resto de las llamadas `exec*()` son frontends de `execve()`).

Esta opción es útil para servidores de cuentas "shell" permitiendo llevar un control de los usuarios.

Si la opción `sysctl` esta habilitada, se creará una entrada dentro de `sysctl` con el nombre `exec_logging`.

**ADVERTENCIA:** Esta opción genera una ingente cantidad de logs, especialmente si es un sistema activo.

- **Resource logging - Registro de recursos**

Activando esta opción, todos los intentos de sobrepasar los límites de recursos quedarán registrados con el nombre del recurso, el tamaño solicitado y el límite actual. Habilitar esta opción es altamente recomendable.

- **Signal logging - Registro de Signals**

Activando esta opción, las señales de cierta importancia serán registradas, como es `SIGSEGV`, que como resultado mostrará cuando un programa ha terminado con un error, que en algunos casos puede significar un intento de exploit.

Si la opción `sysctl` está activada, dentro de `sysctl` aparecerá una opción con el nombre `signal_logging`.

- **Fork failure logging - Registro de fallo de Fork**

Activando esta opción, todos los intentos fallidos de `fork()` serán registrados. Esto puede significar un intento de "bomba fork", o que alguien ha intentado sobrepasar el límite de sus procesos.

Si la opción `sysctl` está activada, se creará dentro de `sysctl` una opción con el nombre `forkfail_logging`.

- **Time change logging - Registro de cambio de hora**

Activando esta opción, cualquier cambio en el reloj del sistema será registrado.

Si la opción `sysctl` está activada, se creará dentro de `sysctl` una opción con el nombre `timechange_logging`.

- **ELF text relocations logging - Registro de las reubicaciones de texto en ELF**

Activando esta opción, las reubicaciones de texto serán registradas junto con el nombre del binario o de la librería que la causa. El propósito de esta funcionalidad es ayudar a los desarrolladores de Linux a identificar las librerías y binarios que necesitan las reubicaciones de texto, y que obstaculizan el progreso de PaX. Únicamente los desarrolladores de Linux deberían de activar esta opción, y nunca en un sistema en producción, ya que se estaría creando una fuga de información que podría permitir evitar la generación aleatoria de una única región de memoria.

Si la opción `sysctl` está activada, se creará dentro de `sysctl` una opción con el nombre `audit_texrel`.

- **Log all execs within chroot - Registrar todos los execs dentro del chroot**

Activando esta opción, se registrarán todas las ejecuciones dentro de la jaula `chroot` en el `syslog`.

Hay que considerar que con determinadas aplicaciones, por ejemplo `djb's daemontools`, esta opción generará una gran cantidad de logs, por esto es opcional.

Si la opción `sysctl` esta activada, se creará una opción dentro de `sysctl` con el nombre `chroot_execlog`.

## EXECUTABLE PROTECTIONS - PROTECCIÓN DE EJECUTABLES

En esta sección se podrán configurar aquellas opciones que intervienen en la creación de procesos y a que binarios del sistema se puede acceder.

- **Enforce RLIMIT\_NPROC on execs**

Activando esta opción, los usuarios con un límite de recursos por proceso tendrán que tener activado el valor durante las llamadas a `execve()`.

El sistema actual únicamente comprueba el límite durante las llamadas a `fork()`.

Si la opción `sysctl` esta activada, se creará una opción dentro de `sysctl` con el nombre `execve_limiting`

- **Trusted path execution - Ruta de ejecución confiable**

Activando esta opción, se podrá seleccionar un `gid` para añadir grupos adicionales de usuarios que se quieran identificar como no confiables `untrusted`.

Estos usuarios no podrán ejecutar ningún fichero que no este en directorios cuyo propietario no sea `root` y que sólo sean de escritura para `root`.

Si la opción `sysctl` esta habilitada, se creará dentro de `sysctl` una opción con el nombre `tpe_gid`.

- **Partially restrict non-root users - Restricciones parciales a usuarios no root**

Activando esta opción, Todos los usuarios distintos de `root` exceptuando los especificados en el grupo TPE (opción anterior) tendrán permisos para ejecutar ficheros que se encuentren en directorios propiedad de `root` y con permisos de escritura solo para `root`.

Si la opción `sysctl` esta habilitada, se creará dentro de `sysctl` una opción con el nombre `tpe_restrict_all`.

- **Randomized PIDs - PIDs Aleatorios**

Activando esta opción, todos los PIDs creados en el sistema serán generados de forma pseudo aleatoria. Esta medida es extremadamente efectiva en conjunto con las restricciones del `/proc` para evitar que se puedan averiguar los pids de los servicios, etc. Los PIDs también se utilizan en algunos casos como parte del sistema de nombres para ficheros temporales, por lo tanto esta opción debería de mantener también estos nombres impredecibles. Además se utiliza código para asegurar que los números de los PID no sean rehusados demasiado rápido.

Si la opción `sysctl` esta activada, se creará dentro de `sysctl` una opción con el nombre `rand_pids`

## NETWORK PROTECTIONS - PROTECCIONES DE RED

En esta sección se configuraran las opciones relacionadas con la selección aleatoria de la pila TCP/IP y restricciones sobre que tipos de sockets podrán utilizar los usuarios.

- **Larger entropy pools - Mayores piscinas de entropia**

Activando esta opción, las piscinas de entropía usadas para muchas de las características de Linux y `grsecurity` duplicarán su tamaño. Considerando que muchas de las funcionalidades de `grsecurity` utilizan aleatoriedad adicional, es recomendable activar esta opción.

Activar esta opción tiene un efecto similar a modificar `/proc/sys/kernel/random/poolsize`.

- **Randomized TCP source ports - Generación aleatoria de puertos de origen TCP**

Activando esta opción, las situaciones donde el puerto de origen es generado en marcha por el protocolo TCP, (por ejemplo. con `connect()`) serán modificadas de forma que el puerto de origen se generará de forma aleatoria, en lugar de emplear un sencillo algoritmo incremental.

Si la opción de `Sysctl` esta activada, se creará dentro de `sysctl` una opción con el nombre `rand_tcp_src_ports`

- **Socket restrictions - Restricciones en Sockets**

Activando esta opción, podrás seleccionar distintas opciones. Si asignas un `GID` en el sistema y se añade a los grupos de usuarios con acceso restringido a sockets, este parche puede actuar de 3 formas distintas en función de las opciones seleccionadas.

- **Deny all socket access - Denegar todo el acceso a sockets**

Activando esta opción, se podrá seleccionar un `GID` cuyos usuarios no podrán conectar a otros `host` desde el sistema o ejecutar aplicaciones servidor en el sistema.

Si la opción `sysctl` esta habilitada, se creará dentro de `sysctl` una opción con el nombre `socket_all`.

- **Group for disabled socket access - Grupo de usuarios sin acceso a sockets**

Aquí se especificará el `GID` para el cual queremos desactivar los sockets. Recuerda añadir a este grupo a aquellos usuarios para los que se quieren desactivar los sockets.

Si la opción `sysctl` se encuentra habilitada, no importa lo que se seleccione aquí. Se tendrá que especificar el `GID` en el script de inicio enviando mediante `echo` el `GID` a la correspondiente entrada del `/proc`. Para más información, consultar la ayuda en `sysctl` sobre esta opción.

Si la opción `sysctl` se encuentra habilitada, se creará dentro de `sysctl` una opción con el nombre `socket_all_gid`.

- **Deny all client socket access - Denegar el acceso a todos los sockets cliente**

Activando esta opción, se podrá seleccionar un `GID` de usuarios que no podrán conectar a otros sistemas desde el sistema local, pero podrán ejecutar aplicaciones servidor. Si esta opción esta habilitada, todos los usuarios incluidos en este grupo tendrán que emplear el modo pasivo para las transferencias `ftp` desde la cuenta local.

Si la opción `sysctl` esta activada, se creará dentro de `sysctl` una opción con el nombre `socket_client`.

- **Group for disabled client socket access - Grupo de usuarios sin acceso a sockets cliente**

Aquí se seleccionará el `GID` cuyos usuarios no podrán acceder a sockets cliente. Recuerda añadir a este grupo a aquellos usuarios para los cuales quieras eliminar el acceso a sockets cliente.

Si la opción `sysctl` esta activada, no importa lo que se seleccione aquí. Se tendrá que especificar el `GID` en el script de inicio enviando mediante `echo` el `GID` a la correspondiente entrada en el `/proc`. Para más información, consultar la ayuda en `sysctl` sobre esta opción.

Si la opción `sysctl` se encuentra activa, se creará dentro de `sysctl` una opción con el nombre `socket_client_gid`.

- **Deny all server socket access - Prohibir el acceso a todos los sockets del servidor**

Activando esta opción, se podrá seleccionar un `GID` cuyos usuarios no podrán ejecutar aplicaciones servidor en el sistema.

Si la opción `sysctl` se encuentra habilitado, se creará dentro de `sysctl` una opción con el nombre `socket_server`.

- **Group for disable server socket access - Grupo de usuarios sin acceso a sockets servidor**

Aquí se seleccionará el `GID` cuyos usuarios tendrán el acceso a sockets de servidor deshabilitado. Recuerda añadir a este grupo a los usuarios a los cuales quieres impedir el acceso a sockets de servidor.

Si la opción `sysctl` se encuentra activada, no importará lo que se seleccione aquí. Se tendrá que especificar el `GID` en el script de inicio enviando mediante `echo` el `GID` a la correspondiente entrada en el `/proc`. Para más información, consultar la ayuda en `sysctl` sobre esta opción.

Si la opción `sysctl` se encuentra activada, se creará dentro de `sysctl` una opción con el nombre `socket_server_gid`.

## **SYSCTL SUPPORT - SOPORTE PARA SYSCTL**

En esta sección se activa el soporte para `sysctl`. Activar esta opción nos permite modificar la configuración para la mayoría de las características de `grsecurity` en tiempo de ejecución. Por defecto esta configurada para desactivar todas las opciones al inicio, por lo tanto no se recomienda activar esta funcionalidad.es de notificación y el número máximo de estos mensajes.

- **Sysctl support - soporte para Sysctl**

Activando esta opción, se podrán cambiar las opciones con las que se ejecuta `Grsecurity` al inicio, sin tener que recompilar el kernel.

Se podrán enviar los valores mediante `echo` a los ficheros ubicados en `/proc/sys/kernel/grsecurity` para activar (1) o desactivar (0) varias opciones. Todas las entradas de `sysctl` son modificables mientras que la entrada `grsec_lock` tenga valor distinto de 0.

Todas las características están desactivadas por defecto. Hay que tener en cuenta que esta opción puede reducir la eficacia de las medidas de seguridad adicionales de este parche si no se usa junto con un sistema de ACL. Los scripts de inicio han de ser de sólo lectura, y `root` no debería de tener acceso a añadir módulos o realizar operaciones de raw i/o. Todas las opciones deben establecerse al inicio, y la entrada `grsec_lock` debería ser configurada a un valor distinto de 0 una vez se han establecido todas las opciones.

**\*ESTO ES EXTREMADAMENTE IMPORTANTE\***

## **LOGGING OPTIONS - OPCIONES DE REGISTROS DE EVENTOS**

En esta sección se permite especificar el tiempo entre mensajes de notificación y el número máximo de estos mensajes.

- **Number of burst messages - Número de mensajes de notificación**

Esta opción permite seleccionar el número máximo de mensajes permitidos dentro del intervalo de `flood` seleccionado en una opción a parte. La configuración por defecto puede ser adecuada para la mayoría de la gente, sin embargo si se interpreta que muchos de los mensajes se están interpretando como `flood`, se deberá incrementar este valor. Por defecto este valor es de 4.

- **Seconds in between log messages - Segundos entre mensajes de registro**

Esta opción permite establecer el número de segundos que han de transcurrir entre cada mensaje de registro de Grsecurity. La configuración por defecto puede ser adecuada para la mayoría de la gente, sin embargo, si se decide cambiar este valor, selecciona un valor suficientemente pequeño para permitir que se produzcan registros de eventos informativos, pero suficientemente grande como para evitar el `flod`. Por defecto este valor es de 10 segundos.

## **ROLE BASED ACCESS CONTROL OPTIONS - OPCIONES DE CONTROL DE ACCESO BASADO EN ROLES**

En esta sección se podrán configurar las opciones relacionadas con el sistema RBAC de Grsecurity, incluyendo el número de intentos de autenticación fallidos que puede realizarse hasta que se produzca el bloqueo de la cuenta por un tiempo determinado.

- **Hide kernel processes - Ocultar los procesos del kernel**

Activando esta opción, cuando el sistema RBAC esta activado a través de `gradm` -una ACL adicional pasará al kernel de forma que ocultara todos los procesos. Estos procesos solo podrán ser vistos por el administrador autenticado, o se verán aquellos procesos que tengan activada la opción de ser vistos.

- **Maximum tries before password lockout - Máximo número de intentos antes del bloqueo de la cuenta.**

Esta opción establece el número máximo de veces que un usuario puede intentar autenticarse en el sistema RBAC antes de impedirle volver a intentar la autenticación por un periodo de tiempo. Cuanto menor sea el número, más complicado será obtener una contraseña mediante un ataque de fuerza bruta.

- **Time to wait after max password tries, in seconds - Tiempo que tendrá que transcurrir despues de superar el máximo número de intentos de autenticación, en segundos**

Esta opción especifica el tiempo que ha de esperar un usuario para volver a autenticarse en el sistema RBAC una vez ha realizado el número máximo de intentos de autenticación. Cuanto mayor sea este número, más complicado será realizar con éxito ataques de fuerza bruta.