UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Trabajo práctico Nro. 1

Shell scripting en Linux

S.H.I.E.L.D

SHELL INCREIBLEMENTE EXIGENTE, LINUXERO Y DIVERTIDO

Ingeniería en Sistemas de Información
Cátedra de Sistemas Operativos

- 2C2012 -

Versión 1.0

Índice

Introducción

El trabajo práctico Nº1 consiste en el desarrollo de una serie de scripts en lenguaje BASH sobre Linux que atacarán problemas específicos de la administración de sistemas operativos.

Objetivos del trabajo práctico

Que el alumno:

- Conozca y comprenda la utilidad de los shell scripts para llevar a cabo la administración de Sistemas operativos.
- Conozca la estructura, sintáxis y semántica de un shell script en Linux.
- Desarrolle la capacidad para resolver problemas específicos de administración de sistemas operativos basados en UNIX.

Desarrollo

Resolución de problemas complejos de administración

Introducción

El proyecto consiste en el desarrollo de un shell (de ahora en más `Shield´) íntegramente realizado mediante shell scripting. Dicho script será ejecutado sobre Bash, siendo este último el shell por defecto asociado a todos los usuarios en sistemas Linux.

Shield tendrá como principales características la posibilidad de incorporar/quitar módulos en forma dinámica, establecer configuraciones particulares a cada usuario y por último, al estar montado sobre Bash, explotar todas las funcionalidades que este ofrece. De esta manera se puede pensar a Shield como un wrapper (envoltorio) de Bash al cual le adiciona diversas funcionalidades.

Flujo de ejecución ejemplificador

A continuación se presenta un ejemplo de uso básico de Shield:

El usuario root descomprime el archivo shield.tar.gz y ejecuta *make instalar*. Luego ejecuta *make configurar* y configura el uso de Shield al usuario *jose*.
José se loguea en el sistema, se le presenta como shell a Shield. En él se registra únicamente el módulo de auditoria.
Pepe ejecuta el comando *"ls /etc/shield | grep shield"*, entonces el núcleo lo procesa con el módulo de auditoria. Este retorna con código de salida exitoso, por lo tanto el núcleo deriva su ejecución a Bash.

En otra terminal se loguea el usuario root y modifica el archivo de configuración de módulos de comando, agregando al usuario pepe el módulo de seguridad.
Luego, el root, crea el archivo de configuración del módulo de seguridad para el usuario pepe y le agrega el comando *"ls"* como comando restringido.
El núcleo luego de un tiempo detecta dicho cambio, entonces decide actualizar los módulos registrados.

José ejecuta nuevamente el comando *"ls /etc/shield | grep shield"*, pero resulta que el módulo de seguridad detecta que es un comando restringido entonces retorna un código de error. Dicha situación hace que el núcleo informe en pantalla el error y no permita al usuario ejecutar el comando.

Componentes de Shield

Núcleo o script principal

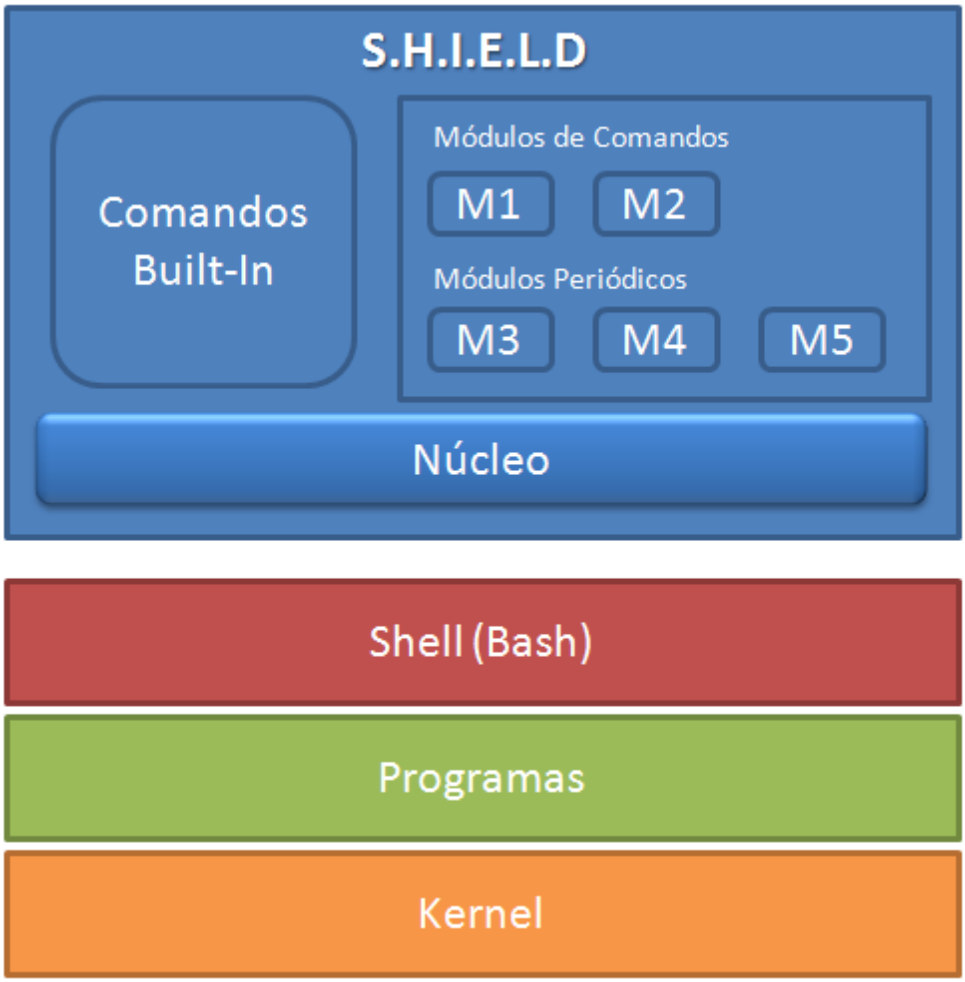
Este componente es el centro o corazón de Shield, sobre él se registrarán *módulos periódicos* y *módulos de comando*. Será el encargado de leer los comandos tipeados por el usuario y procesarlos a través de los módulos registrados.

Este componente será invocado automáticamente por el sistema operativo al inicio de la sesión del usuario[1]. Luego de iniciado, llevará a cabo la función ***Registrar e inicializar módulos*** descripta más adelante. Por último, permanecerá en ejecución a la espera de comandos del usuario.

El núcleo, al igual que Bash, ofrecerá una serie de comandos built-in. Para conocer cuáles son y una descripción detallada de los mismos dirigirse a ***Comandos Built-in de Shield***.

A continuación se presenta un diagrama de la arquitectura del sistema (en azul la parte a desarrollar):

[1] Sólo a aquellos usuarios a los cuáles se les activó el uso de Shield. Para más detalle ver ***Makefile***, el target instalar / configurar.



Las principales funciones de este script[2] serán:

- **Registrar e inicializar módulos:**
  Como primera acción, el núcleo deberá verificar si existen actualmente módulos ejecutando[3], en caso de haberlos, los detendrá.
  Luego, realizará la lectura de los archivos de configuración de los módulos[4] relativos al usuario logueado, y procederá a registrar[5] e inicializar los módulos que se encuentren activados.
  *Aclaración importante: Si al iniciar un módulo, éste retorna un código de salida de **error** deberá informarse tanto en pantalla como en un archivo log[6] dicha situación y luego desloguear al usuario, finalizando de esta manera el shell asociado a esa sesión.*

  Para más información acerca de los archivos de configuración leer la sección ***Archivos de configuración***.

  Para más información acerca de las operaciones factibles sobre los módulos leer la sección ***Módulos***.

- **Verificar periódicamente el cambio de los archivos de configuración de módulos:**
  El núcleo, cada cierto tiempo[7], deberá detectar si se produjo un cambio en los archivos de configuración de los módulos[4] desde la última vez que se accedió a ellos.
  En caso de haberse producido, deberá informarlo por pantalla y llevar a cabo la función ***Registrar e inicializar módulos***.

---

[2] No necesariamente debe ser implementado como un único script. Esto significa que puede ser desagregado en varios scripts y/o bibliotecas de funciones.
[3] Sólo podrán existir módulos ejecutando cuando se acceda a ésta mediante la función ***Verificar periódicamente el cambio de los archivos de configuración de módulos***.
[4] Por archivos de configuración de los módulos se hace referencia a aquellos que definen la lista de módulos de Shield, tanto periódicos como de comando.
[5] La registración de los módulos se realizará en el mismo orden en el que figura en el archivo de configuración.
[6] El archivo de log residirá en la carpeta de shield del usuario: /home/jose/.shield/shell.log

Esto permite que un usuario con permisos de escritura sobre dichos archivos pueda modificar en tiempo de ejecución los módulos registrados en Shield.

- **Administrar la ejecución de los módulos periódicos:**
  El núcleo cada cierto período de tiempo[7] deberá ejecutar en forma secuencial los denominados *módulos periódicos*.
  Aclaración: Secuencial hace referencia a que deben ejecutarse en el mismo orden en que se encuentran registrados, que a su vez es el orden en el que figuran en el archivo de configuración de módulos periódicos.

  En caso que la ejecución de alguno de ellos genere como resultado un código de error, el núcleo deberá informarlo tanto por pantalla como en un archivo de log y desloguear al usuario actual, finalizando de esta manera el shell asociado a esa sesión.

- **Leer y procesar los comandos tipeados por el usuario:**
  El núcleo, al igual que cualquier shell, se encontrará permanentemente a la espera de comandos provenientes del usuario.
  Ante el ingreso de uno, el núcleo ejecutará en forma secuencial los denominados *módulos de comando* enviándoles el `string´ tipeado por el usuario para que estos lo procesen.

  Cómo se detalla más adelante, los módulos retornan un código de salida. Para el núcleo son de particular interés los siguientes:
  - Código de salida de error: Este le indica que el comando tipeado por el usuario produjo resultados negativos en el módulo. En ese caso, el núcleo cortará la ejecución de los siguientes e informará por pantalla dicha situación.
    Esto significa que si por ejemplo en el núcleo están registrados los módulos de comando m1, m2 y m3, al retornar con código de error el m2, no se continuará ejecutando al restante m3.
  - Código de salida exitoso: El procesamiento del comando fue correcto. Entonces se continúa ejecutando el siguiente módulo.

  Cuando todos los módulos se ejecutaron exitosamente, el núcleo procederá a invocar el comando del usuario en el shell (bash). En caso de tratarse de un built-in, lo procesará internamente en el script.
  Ejemplo: El usuario tipea "ls", dicho comando produce resultados exitosos en cada uno de los módulos registrados, entonces shield ejecuta "ls" sobre Bash. Esto hace que se imprima en pantalla el listado de archivos en el directorio actual.

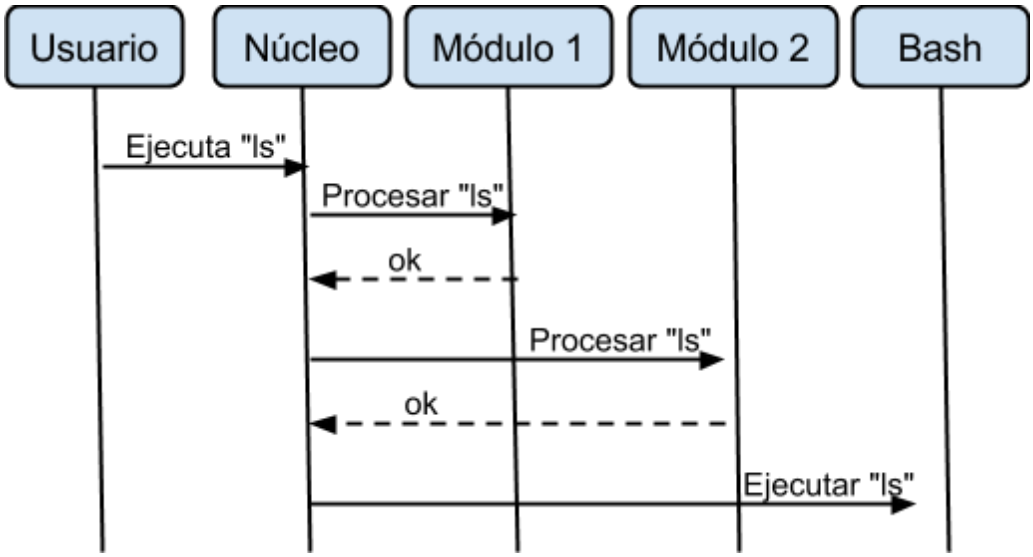
  Aclaración importante: Shield deberá permitir ejecutar comandos que reciban parámetros, redirecciones de E/S, y/o que sean ejecutados en modo background, tal como lo hace posible Bash.
  Ejemplo:

```
[Shield] guest@localhost:~# find / -name "*shield*" &
```

Diagrama de secuencia de procesamiento de comandos

A continuación se presenta un diagrama que expresa mediante un ejemplo la interacción entre los componentes de Shield ante el ingreso de un comando del usuario.

[7] Deberá poder ser configurable mediante algún archivo de configuración definido por el grupo o dentro del mismo script a través de una variable.



Módulos

Los módulos son componentes implementados como shell scripts que proveen las funcionalidades específicas de Shield. Estos pueden ser incorporados/quitados del sistema en forma dinámica mediante la simple edición de los archivos de configuración correspondientes (Para más detalle leer ***Archivos de configuración***).

Una característica importante de los módulos es que respetan una cierta interfaz[8]. Esto le permite al núcleo el poder tratarlos a todos por igual sin necesidad de conocer qué módulo específico tiene registrado.

Por lo tanto, la forma de invocar a cualquier módulo es la siguiente:

<nombreModulo> <tipoOperacion> [<parametro>] [9]

Donde <nombreModulo> es el path absoluto del script ejecutable del módulo, <tipoOperacion> es una cadena que indica el tipo de operación que se quiere llevar a cabo sobre el módulo y <parametro> representa a los argumentos que recibirá el módulo.

A continuación se especifica la interfaz mínima[10] de un módulo:

Tipo de operación	Descripción
informacion	Deberá imprimir en la salida estándar los datos específicos del mismo. Para conocer qué información debe presentar cada módulo, leer la sección ***Especificación detallada de módulos***.
iniciar	Deberá leer su archivo de configuración y setear/exportar las variables de entorno que se desprendan del config con su correspondiente valor. Esto significa que el módulo no leerá el archivo de configuración cada vez que ejecute la operación "procesar", sino que los valores de configuración los tomará de las variables de entorno previamente iniciadas.
detener	Eliminará todas las variables de entorno inicializadas al ejecutar la operación "iniciar". También, si el grupo lo considera necesario, en esta operación se podrá hacer algún tipo de limpieza de archivos u otros.
procesar	Realizará el procesamiento específico del módulo. Para conocer qué proceso debe hacer cada módulo leer ***Especificación detallada de módulos***.

Una característica común a todos los tipos de operación es que retornan un código de salida[11]. Este código podrá tomar diversos valores que dependerán básicamente del éxito o fracaso del procesamiento realizado.

Por ejemplo, si existe un módulo que no permite al usuario ejecutar el comando "ls" y el usuario ingresa dicho comando en Shield, el módulo, al ser invocado con el tipoOperacion igual a

[8] Para investigar en mayor detalle cómo se implementa una interfaz común entre diversos scripts, analizar **exhaustivamente** los scripts ubicados en el path "/etc/init.d". ¿Qué representan estos scripts?, ¿Para qué y cómo se los utiliza?.

[9] Los corchetes indican que el parámetro es opcional. Esto dependerá del tipo de módulo que se haya ejecutado. Para más información leer las secciones ***Módulos de comando*** y ***Módulos periódicos***.

[10] Se indica mínima porque el grupo puede agregar otros tipos de operaciones.

[11] Los valores de los códigos de salida de un módulo podrán ser elegidos a gusto del grupo (Siempre y cuando permitan distinguir una condición de otra). Adicionalmente, el grupo, si así lo considera, podrá agregar otros tipos de código de salida aparte del de *éxito* y *error*.

*procesar, generará un código de salida de **error**, en caso contrario de **éxito.***

Se sugiere una organización jerárquica de directorios que distribuya los módulos en carpetas separadas. Por ejemplo, suponiendo que el directorio de instalación fuera /etc/shield, la ruta de un módulo podría ser:

```
/etc/shield/modulos/comando/seguridad/modulo_seguridad.sh
```

Existen dos tipos básicos de módulo, los de comando y los periódicos. A continuación se explican ambos en detalle:

Módulos de comando

Son aquellos que se encargan de procesar comandos tipeados por el usuario. Estos módulos serán invocados por el núcleo ante el ingreso de cada comando.

Como se aclaró anteriormente, la forma de invocar a un módulo es la siguiente:

<nombreModulo> <tipoOperacion> [<parametro>]

En estos módulos, <parametro> hace referencia al comando tipeado por el usuario. De esta manera, si se ingresa "ls | grep pepe" la invocación al módulo sería por ejemplo:

/etc/shield/modulos/moduloX/moduloX.sh procesar "ls | grep pepe".

Módulos periódicos

Se denominan así a los módulos que serán invocados periódicamente por el núcleo. A estos módulos no necesariamente se les deberá enviar <parametro>[12].

Archivos de configuración (config)

En esta sección se presentarán los archivos de configuración asociados a los componentes de Shield[13]. Todos los archivos de configuración serán "por usuario" (con Shield configurado), y residirán en una carpeta oculta en la home de dicho usuario.

Configs de Módulos habilitados por usuario

El núcleo tendrá, por cada usuario con Shield configurado, dos archivos de configuración, uno con la lista de módulos de comando y otro con la lista de módulos periódicos.

La idea de estos archivos es que declaren los módulos de los cuales dispone Shield para un usuario específico.

Ambos archivos tendrán la misma sintaxis, a saber:

Por cada módulo se creará una línea con el path absoluto del mismo, luego un ":" (dos puntos) como separador y luego la palabra on/off para indicar si el módulo se encuentra activado o no.

Ejemplo: Dados dos módulos de comando y uno periódico configurados para el usuario jose, el contenido de los archivos puede ser el siguiente:

Config de módulos de comando

```
/etc/shield/modulos/comando/seguridad/modulo_seguridad.sh:on
/etc/shield/modulos/comando/auditoria/modulo_auditoria.sh:off
```

Config de módulos periódicos

```
/etc/shield/modulos/periodicos/limitacion/modulo_limitacion.sh:on
```

Configs de los módulos

Cada módulo de Shield tendrá por cada usuario un archivo de configuración. *Su contenido dependerá de cada módulo y sus requerimientos. Para más detalle ver **Especificación detallada de módulos**.*

Makefile

Shield utilizará como herramienta de instalación/desinstalación y configuración de usuarios, la aplicación make.
El grupo, por lo tanto, deberá desarrollar un archivo makefile que contenga mínimamente los siguientes targets:

[12] Si el grupo lo considera necesario podrá hacerlo.
[13] Si el grupo lo desea puede agregar más archivos de configuración que los presentados en este documento.

Target	Descripción
instalar	Se distribuirán todos los scripts (nucleo principal, y scripts utilitarios y scripts de modulos) a una carpeta específica del sistema, por ejemplo: /etc/shield (la misma debería poder ser configurada, aunque sea modificando una variable del script antes de instalar). Se deberá crear un symbolic link en la ruta **/usr/bin/shield.sh**, que deberá apuntar al script principal de shield (instalado en la carpeta especificada anteriormente). No deberá permitir instalar a shield si ya se encuentra instalado en el sistema
desinstalar	Se eliminarán todos los scripts asociados a Shield (ignorando los archivos de configuración de los usuarios) No deberá permitir desinstalar si shield no se encuentra instalado o si existe algún usuario que todavía lo tiene configurado.
configurar	Permite configurar el uso de Shield como shell por defecto a un usuario específico. De esta manera cuando dicho usuario se loguee, el sistema operativo le presentará automáticamente a Shield en lugar de Bash. La configuración del shell será referenciando al symbolic link creado en la instalación. Se deberá crear una carpeta oculta en la home del usuario que contenga toda la configuración específica de Shield. Todos los directorios dentro de esta carpeta (incluída la misma), y todos los archivos de configuración; deberán tener como propietario al usuario root, y el resto de los usuarios deberán tener solamente permisos de lectura. No deberá permitir configurar si shield no se encuentra instalado o si el usuario elegido ya lo tiene configurado.
resetear	Permite des-configurar a Shield de un usuario, eliminando todos sus archivos de configuración y reinstaurando al usuario el shell que previamente tenía configurado. No deberá permitir resetear si shield no se encuentra configurado para el usuario.

Sólo el usuario root podrá ejecutar los targets del makefile. En caso de intentar acceder con otro usuario, informar por pantalla dicha situación y salir.

Se podrán agregar nuevos targets, a medida que el grupo lo considere necesario, como por ejemplo reinstalar, reconfigurar, etc.

La instalación y/o configuración de Shield para un usuario no debería impedirle al mismo poder luego cambiarse a otro shell por sus propios medios, e inclusive luego volver a cambiar por Shield considerándolo un shell más (siempre y cuando no haya sido reseteado para ese usuario)

Comandos Built-in de Shield

A continuación se presenta una lista de comandos embebidos en Shield:

Comando	Descripción
ayuda [builtin]	Presenta en pantalla una breve ayuda el builtin indicado. Si no se recibe parámetro, presentará la ayuda de todos los builtins.
info_modulos [string]	A cada módulo activo para el usuario le indica que imprima en pantalla información sobre sí mismo. *Para más detalle sobre las operaciones factibles sobre un módulo leer **Módulos**.* Si al built-in se lo invoca en la forma "*info_modulos string"*, sólo imprimirá la información de los modulos cuyo nombre contengan al string recibido.
listar_modulos	Presenta en pantalla los paths absolutos de los módulos que tiene activo el usuario.
actualizar_modulos	Invoca a la función del núcleo ***Registrar e inicializar módulos***.
mostrar variable1	Muestra el valor de la variable interna del shell de nombre "variable1"
salir	Termina la sesión actual del usuario.
apagar	Apaga la pc (deberá hacerlo sin solicitar al usuario ningún tipo de password)

Especificación detallada de módulos

En esta sección se explicarán en forma detallada cada uno de los módulos que deberán desarrollarse para Shield.[14]

Módulos de comando

Módulo de seguridad

Se encargará de restringir la ejecución de ciertos comandos al usuario. Dichos

[14] Adicionalmente, durante la evaluación del TP, podrán proporcionarse al grupo nuevos módulos desarrollados por la cátedra que deberán integrarse sin problemas a Shield. Deberá ser suficiente, para la instalación de un nuevo módulo, el copiado de los scripts y configs del nuevo módulo en la carpeta indicada por el grupo.

comandos figurarán en el archivo de configuración del módulo (Recordar que hay uno por usuario).

Ejemplo: Dado un usuario jose y el siguiente contenido de su config para el módulo de seguridad:

ssh
scp
telnet

El módulo producirá los siguientes códigos de salida para los comandos tipeados:

"ls" → éxito
"ls | grep hola" → éxito
"ssh 192.168.2.10" → error
"echo Hola | telnet 192.168.2.10" → error

Al invocar al módulo con el <tipoOperacion> igual a información, deberá imprimir en pantalla los comandos restringidos al usuario.

Módulo de auditoria

Se encargará de loguear en un archivo los comandos completos (incluyendo parámetros, pipelines y/o redireccioines usados) tipeados por el usuario. El archivo residirá en la carpeta oculta del shell en el home del usuario.

Parámetros configurables del módulo	**Descripción**
Tamaño máximo de archivo local de log	Indica el tamaño máximo admitido para el archivo de log local de Shield. Cuando el mismo supere dicho valor, todo logueo deberá realizarse en forma remota sobre el servidor indicado por el parámetro *IP del servidor de log.* *Aclaración I: La unidad de tamaño será en bytes.* *Aclaración II: El logueo remoto deberá ser realizado sobre un archivo que resida en el home del usuario de mismo nombre al usuario local. Ejemplo, si el usuario local es jose, deberá loguear en el path /home/jose del servidor remoto. (Se asume la existencia de dicho usuario en el servidor de log, en caso de no existir se creará manualmente).*
IP del servidor de log	IP del servidor sobre el cual se realizará el logueo en caso de superar el tamaño máximo de archivo de log local.

El módulo retornará siempre un código de salida exitoso (Salvo que el grupo decida crear otros códigos para diferenciar situaciones).

Al invocar al módulo con el <tipoOperacion> igual a *información*, deberá imprimir en pantalla los valores de los parámetros de configuración y el tamaño actual del archivo local de log.

Módulo de control de sesiones

Se encargará de limitar la cantidad de sesiones de Shield abiertas por el usuario.
Este módulo, a diferencia de los explicados anteriormente, no realizará ninguna operación cuando se lo invoque con el <tipoOperacion> igual a *procesar*.
El control de sesiones lo realizará cuando se inicie el módulo, es decir, al invocarse con el <tipoOperacion> igual a *iniciar*.
De esta manera, se controlará la cantidad de sesiones abiertas por el usuario al momento de cargar el módulo en Shield.

Como se explicó en la sección ***Módulos***, se retornará un código de salida de error si la cantidad de sesiones supera la máxima establecida. Dicho valor deberá ser leído del archivo de configuración.

Al invocar al módulo con el <tipoOperacion> igual a *información*, deberá imprimir en pantalla los valores de los parámetros de configuración y la cantidad actual de sesiones abiertas por el usuario.

Módulos periódicos

Módulo de limitaciones

Establecerá limitaciones respecto al consumo de recursos dentro de una sesión de

Shield.[15] A continuación se presentan los parámetros de configuración y un detalle de las limitaciones que deben realizarse.

Parámetros configurables del módulo	
Máximo uso de CPU por sesión	Es un valor porcentual que representa el máximo consumo de CPU que puede realizar el usuario en una terminal de Shield. El uso de CPU por sesión se obtiene como sumatoria de todos los consumos de CPU de los procesos activos[16] iniciados en la sesión.
Máximo uso de Memoria por sesión	Al igual que con el máximo uso de CPU, con la diferencia que tiene en cuenta el consumo de memoria real de todos los procesos activos iniciados en la sesión.
Máximo de procesos por sesión	Valor entero que indica la cantidad máxima de procesos activos que puede tener un usuario en una sesión de Shield.
Máximo de sockets por sesión	Valor entero que indica la cantidad máxima de sockets abiertos por procesos activos iniciados en la sesión. Esta operación deberá ser llevada a cabo mediante el acceso al directorio **/proc**. Para más información *man proc*.
Máximo de archivos abiertos por sesión	Valor entero que indica la cantidad máxima de archivos abiertos por procesos activos iniciados en la sesión.

El módulo producirá un código de salida erróneo si alguna de las limitaciones es excedida.

Al invocar al módulo con el <tipoOperacion> igual a *información*, deberá imprimir en pantalla los valores de los parámetros de configuración y el valor actual de cada una de las restricciones.

Módulo de control de carga

Realizará un seguimiento del consumo de CPU de los procesos asociados a una sesión de Shield, intentando mantenerlo debajo de un determinado nivel.

Más en detalle, dado un nivel máximo de consumo de CPU por proceso obtenido del config, el módulo seleccionará al primer proceso que supere (en mayor medida) dicho consumo y luego le incrementará en 5 unidades su valor de nice.

Cuando un proceso sufra de manera continua 4 incrementos del nice, el módulo lo eliminará del sistema e imprimirá en pantalla dicha acción.

El módulo producirá siempre un código de salida exitoso.

Al invocar al módulo con el <tipoOperacion> igual a *información*, deberá imprimir en pantalla los valores de los parámetros de configuración, el proceso, su nice actual y la cantidad de incrementos de nice del mismo (en caso de existir).

Módulo de limitación de tráfico de red

Intentará establecer una limitación a la cantidad de paquetes IP salientes del nodo de red.

Más en detalle, cuando el módulo detecte que la cantidad de paquetes IP salientes del nodo supera el valor máximo (obtenido del config), se realizará lo siguiente:
Por cada proceso activo iniciado en la sesión actual que posea uno o más sockets con conexiones externas[17], se informarán por pantalla los sockets y luego se procederá a eliminar al proceso del sistema.
De esta manera se penalizan a aquellos procesos del usuario de la sesión actual que puedan llegar a producir tráfico en la red.

El módulo producirá siempre un código de salida exitoso.

Al invocar al módulo con el <tipoOperacion> igual a *información*, deberá imprimir en pantalla los valores de los parámetros de configuración y la cantidad actual de paquetes IP salientes.

[15] Por sesión se hace referencia a la relación que se establece entre el usuario y el sistema operativo tras el logueo. En una misma PC pueden existir múltiples sesiones.
[16] Procesos actualmente en ejecución, bloqueados o listos para ejecutar.
[17] Es decir, sockets conectados con otros nodos de red, no con localhost.