Notas Técnicas UHB

# Índice de Contenidos

1 Índice de Contenidos 1

2 Entorno Inicial del Proyecto 2

2.1 GitHub 2

2.2 Máquinas Virtuales utilizadas con VirtualBox 2

3 Estructura Inicial de UHB 4

# Entorno Inicial del Proyecto

El entorno utilizado en el desarrollo del *Unified Hardening Binary* (UHB) es la siguiente:

1. GitHub como plataforma de desarrollo colectivo y el alojamiento del proyecto.
2. Hipervisor Oracle VirtualBox BM para la creación de las máquinas virtuales de FreeBSD y Debian.
3. Visual Studio Code para los scripts en *bash* y los programas en C, C++.

## GitHub

Se ha creado un repositorio de GitHub para albergar el proyecto y tener un control de las distintas versiones de UHB que se crearán. El repositorio será privado hasta que se termine por completo el trabajo.

También se otorgará acceso de visualización del repositorio al profesor escogido para que pueda realizar seguimientos periódicos del estado del trabajo.

## Máquinas Virtuales utilizadas con VirtualBox

Se utilizarán 3 máquinas virtuales en este trabajo:

1. Dos MVs en el cual se desarrollará y testeará el UHB:
   1. Un FreeBSD 14.2 STABLE, llamado “BSD”
   2. Un Debian 12.9.0, llamado “DEB”
2. Adicionalmente, se dispondrá de una tercera MV, también Debian, llamado de “DESKTOP”. Éste servirá como intermediario entre BSD, DEB y el host para la transferencia de archivos por medio de SSH y las conexiones de red.

Ambos DEB y BSD tendrán las siguientes características:

* 4 GB de RAM
* 2 procesadores
* 8 GB de memoria en una única partición, dividido en:
  + 7 GB de memoria, GPT para BSD, ext4 para DEB
  + 1 GB de memoria *swap*
* 2 adaptadores de red:
  + Un adaptador NAT para la conexión de la máquina con la red, para la descarga de los paquetes necesarios.
  + Un adaptador de red interna para la comunicación entre BSD, DEB y DESKTOP.

Ambos BSD y DEB utilizarán su configuración de instalación por defecto, salvo pequeñas excepciones para DEB para no utilizar el entorno gráfico GNOME y la instalación del servidor SSH.

La MV DESKTOP sí tendrá un entorno gráfico para adicional comodidad. También utilizará el plugin de Oracle VirtualBox Guest Additions para la transferencia de archivos entre DESKTOP y el host.

# Estructura Inicial de UHB

La estructura inicial del proyecto será la siguiente:

1. Un fichero de código fuente llamado *source*, con los siguientes archivos .c:
   1. main.c: Punto de entrada del programa.
   2. so\_detect.c: Implementa las funciones para detectar el sistema operativo y usuario utilizados.
   3. imp\_bsd.c: Implementación de UHB para FreeBSD.
   4. imp\_deb.c: Implementación de UHB para Debian.
   5. utils.c: Archivo de funciones auxiliares, caso necesario.
2. Un fichero de archivos de cabecera, llamado *include*.
3. Carpeta del ejecutable, llamado *bin*, dónde se almacena el Makefile y el binario al compilarlo.
4. Una carpeta llamada *config*, dónde se almacenarán las configuraciones generadas por uhb:
   1. *uhb\_config\_ini*: Configuración inicial al ejecutarse UHB.
   2. *uhb\_config\_(timestamp)*: Configuración generada por el usuario con los cambios realizados mediante UHB.

La estructura de UHB está diseñada de forma modular, de forma que pueda adaptarse a las peculiaridades y diferencias de cada distribución Linux o BSD mediante sus archivos de implementación.

## Flujo del Programa

El flujo de funcionamiento de UHB será la siguiente:

1. *main.c* detectará si el usuario que está ejecutando UHB es root. Caso lo sea, verificará si el SO es soportado mediante *so\_detect.c*.
2. *so\_detect.c* devolverá el sistema operativo utilizado y ejecutará la implementación concreta (*imp\_bsd.c*, *imp\_deb.c*…etc.) de UHB.
3. Una vez dentro de la implementación concreta, el binario efectuará las siguientes búsquedas:
   1. Verificará y recogerá información de las particiones existentes y su tipo mediante *fstab*.
   2. Mirará la existencia y los contenidos de *rc.local* para determinar si se ejecutan comandos pertinentes a permisos en *boot-time*.
   3. Detectará la existencia o no de los programas necesarios soportados para las tareas de endurecimiento.
   4. Se volcará la información de 3.1-3. a *uhb\_config\_ini*.
4. Tras hacer las comprobaciones iniciales, se proporcionará al usuario un menú donde tendrá las opciones de endurecimiento del sistema a escoger.
5. Si el usuario elige terminar, se generará el archivo de configuración y se terminará el programa.

## Funcionalidades de UHB

El usuario dispondrá de las siguientes opciones en el menú:

1. Editar permisos DAC y de los ACLs en el sistema.
2. Editar permisos MAC del sistema.
3. Editar el firewall.
4. Visualizar la configuración actual.
5. Generar configuración a partir de un archivo de configuración existente.
6. Generar archivo de configuración actual y terminar el programa.
7. Terminar el programa sin generar un archivo de configuración.