Fuente: IA Stable Diffusion

LABORATORIO 00B. INFRAESTRUCTURAS AUTOMATIZADAS DE SSI

DEPARTAMENTO DE INFORMÁTICA. UNIVERSIDAD DE OVIEDO

Seguridad de Sistemas Informáticos | 2022 – 2023 (V3.1 *"S-81 Isaac Peral"*)

CONTENIDO

AVISO

Este material forma parte de la asignatura "*Seguridad de Sistemas Informáticos*", impartida en la *Escuela de Ingeniería Informática* de la *Universidad de Oviedo*. Es fruto del trabajo continuado de elaboración, soporte, mejora, actualización y revisión del siguiente equipo de profesores desde el año 2019

- Enrique Juan de Andrés Galiana
- Fernando Cano Espinosa
- Miguel Riesco Albizu
- José Manuel Redondo López
- Luís Vinuesa Martínez

Te pedimos por favor que **NO lo compartas públicamente en Internet**. No obstante, entendemos que puedas considerar este material interesante para otras personas. Por ese motivo, hemos creado una versión de este adaptada para que pueda cursarse de forma online, disponible gratuitamente para todo el mundo y que puedes encontrar en esta dirección: https://ocw.uniovi.es/course/view.php?id=109

A diferencia de esta versión, **la versión libre puedes promocionarla todo lo que quieras**, que para eso está 😊

GRACIAS POR TU COLABORACIÓN

¿Por qué *Docker*?

Docker se utiliza en esta asignatura curso como una tecnología de aislamiento de procesos que permite emular infraestructuras de máquinas a usar en los laboratorios, obteniendo las siguientes ventajas:

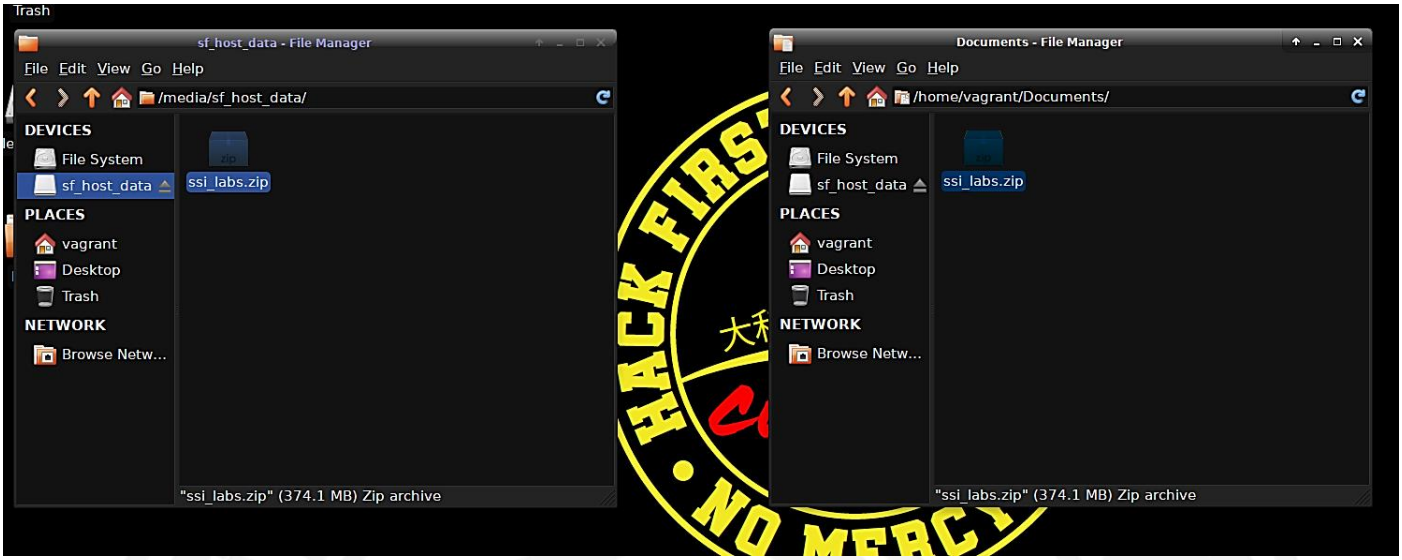
- **Puedes desplegar estas infraestructuras en casa fácilmente**. Simplemente copia los archivos que te proporcionamos y **ejecuta los *scripts* apropiados** (`prepare_labX.sh` y `bulid_labX.sh`, en ese orden) para construir cada infraestructura.
- **La construcción de infraestructuras solo debe hacerse una vez**. Al intentar construirlas de nuevo detectará que ya está construida y omitirá la mayoría (o todos) los pasos.
- **Estos tipos de infraestructuras utilizan muchos menos recursos que máquinas virtuales equivalentes**. No son máquinas virtuales, pero están preparadas para funcionar como si lo fueran, y la mayoría de nuestros laboratorios realmente no requieren una máquina virtual real para hacerse.
- *Docker* está disponible en *Windows, Linux* y *MAC*. Las pruebas se han realizado en nuestra máquina virtual *Ubuntu Server 18.04,* pero nada debería impedirte desplegar la misma infraestructura en diferentes sistemas operativos que admitan *Docker* y *Docker Compose* (incluso de forma nativa). **¡Los laboratorios pueden ser multiplataforma! (NOTA: No hemos podido probar si esta infraestructura funciona con una instalación nativa de *Docker* y *Docker Compose* en *Windows* y *Mac,* por lo que no lo recomendamos).** ¿Por qué lo hacemos todo bajo una misma máquina virtual? Para que todos partáis de la misma base conocida, con el mismo software**.**
- La infraestructura que construirás se basa en archivos de texto e **imágenes oficiales del *Docker Hub.*** Esto significa que los bloques de construcción son "oficiales", y las instrucciones son simplemente archivos de texto. Si no es porque también te proporcionamos clones de sitios web reales, la especificación de las infraestructuras solo debería ocupar unos pocos **Kb.**
- Sí, **todas las infraestructuras funcionan de manera eficiente dentro de nuestra máquina virtual *Ubuntu.*** Como hemos dicho, ¡NO son máquinas virtuales! 😊
- **No necesitas entender nada más que estas instrucciones mínimas sobre *Docker* y *Docker Compose*** para trabajar con ellas. Todo está diseñado para funcionar automáticamente a través de nuestros *scripts*.
- **Todos los contenedores *Docker* que suministramos con la capacidad de entrar en sesión en ellos están configurados como terminales interactivos enriquecidos:** vienen con `tmux`, integración de ratón (módulo `gpm`), navegador de archivos gráfico basado en texto (*Midnight Commander,* `mc`) y una configuración `nano` personalizada avanzada para editar mejor los archivos. Esto tiene como objetivo aumentar tu productividad cuando trabajas dentro de contenedores.

NOTA: Los contenedores *Docker* son **entornos efímeros.** Esto significa que no están preparados para instalar más software que el que te proporcionamos dentro de cada uno de ellos. Sin embargo, en caso de que necesites software adicional por alguna razón, puedes hacerlo como en una instalación normal de *Ubuntu*. Sin embargo, ten en cuenta que, **una vez que salgas de cada laboratorio, los cambios que hagas en sus contenedores se perderán.**

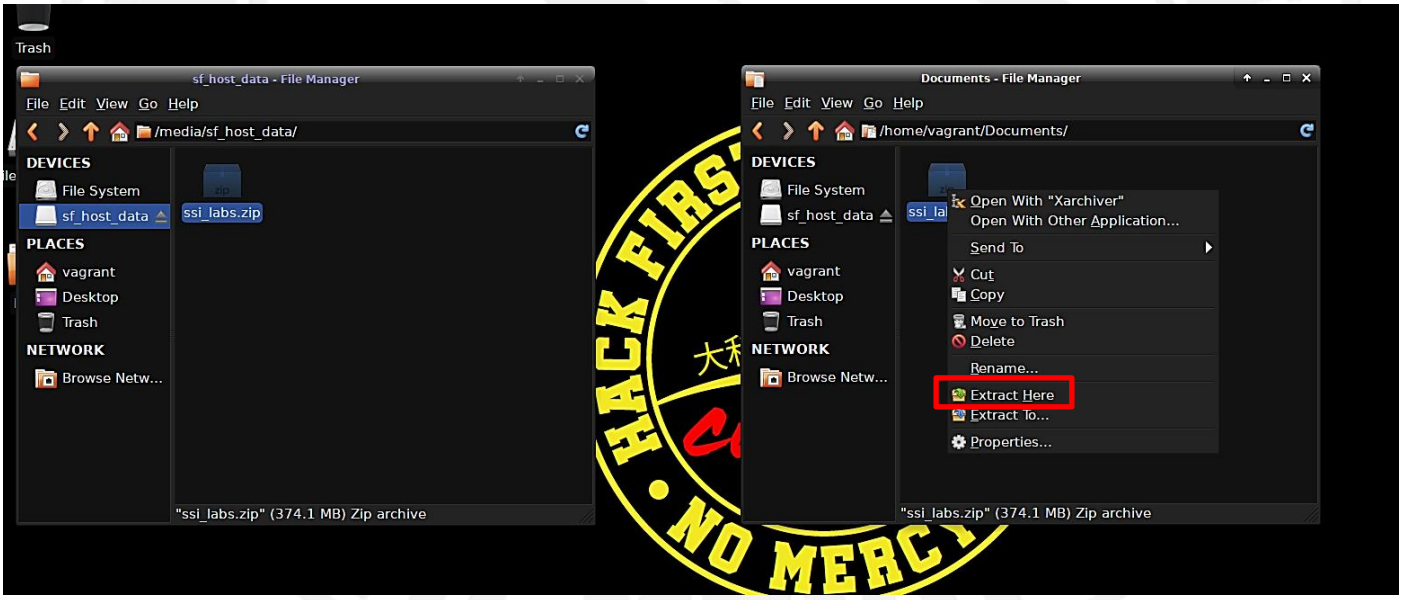
¿Cómo trabajamos con las infraestructuras de los laboratorios?

Lo primero que debes hacer es descargarte el archivo de las infraestructuras de todos los laboratorios. Para ello, copia el archivo `ssi_labs.zip` a una carpeta de la máquina virtual. Si utilizaste la opción 1 para construir la máquina virtual, **este archivo ya está en la carpeta `sf_host_data` de la misma** (la que se usa para compartir

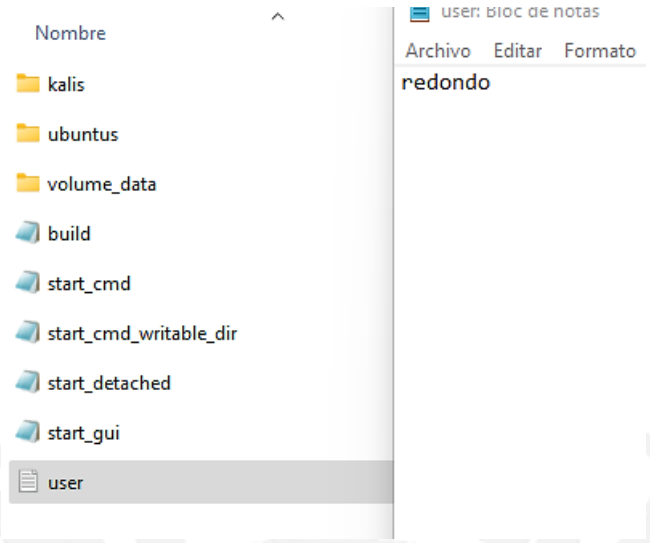
contenido con tu equipo). Si usaste la opción 2, el fichero estará en el escritorio de tu usuario de inicio de sesión. Si creaste tu propia máquina virtual (opción 3), debes descargar el archivo desde el campus virtual.



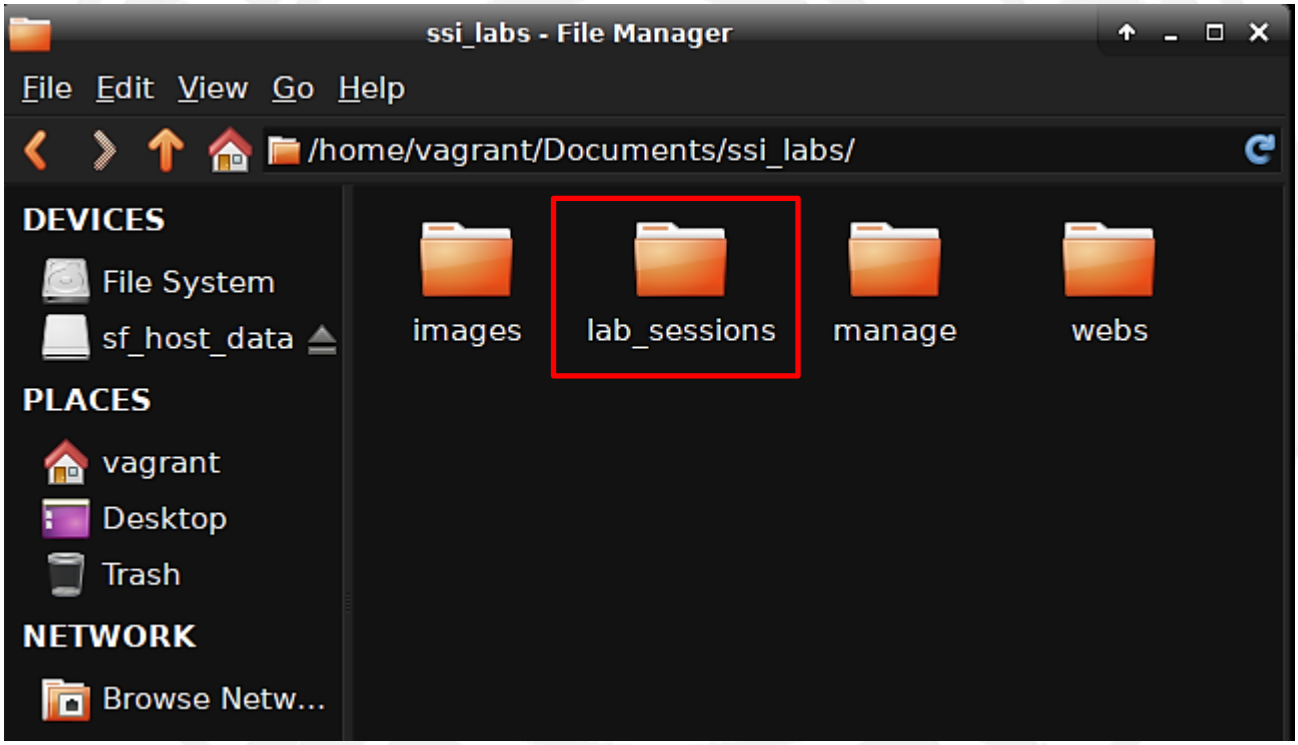
Una vez que copies el archivo, descomprímelo. La GUI tiene una opción gráfica para hacerlo. También puedes usar la herramienta de línea de comandos `unzip`.



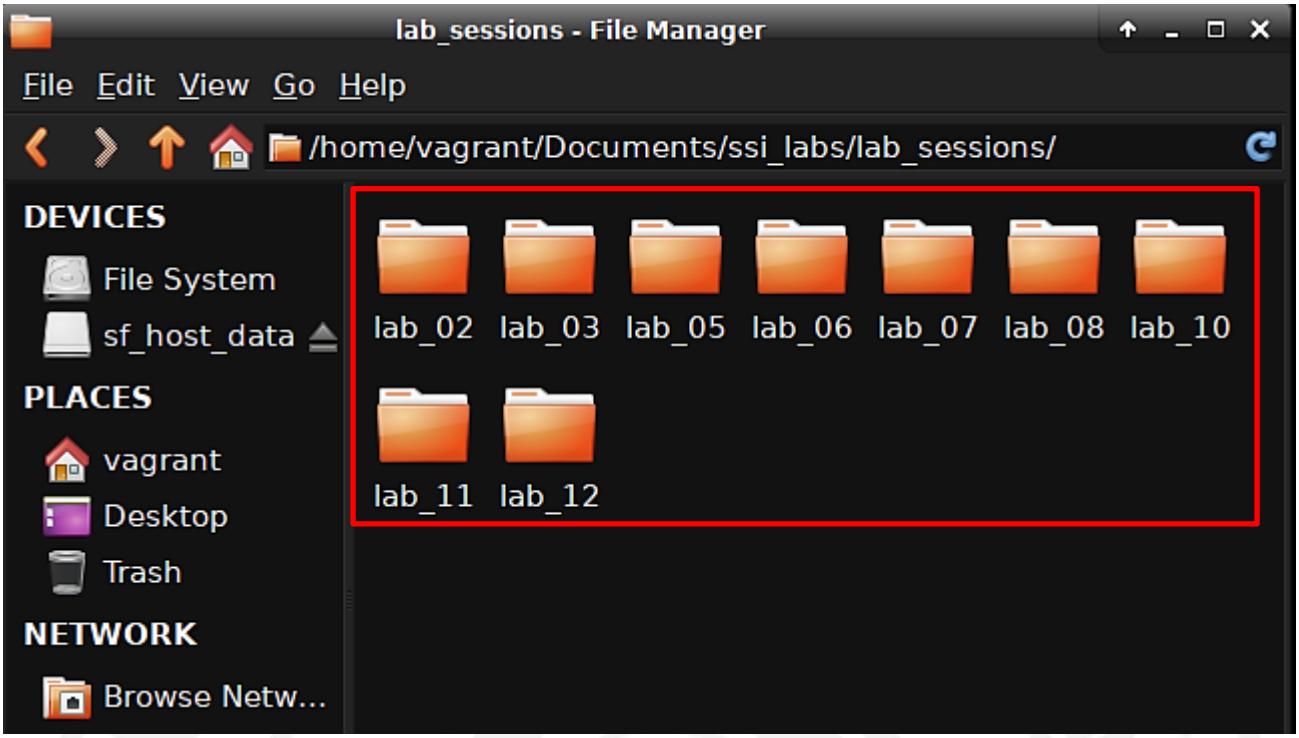
Una vez has descomprimido todo, vete a la carpeta `ssi_labs/images` y (si quieres) abre el fichero `user.txt`. **Aquí puedes cambiar el nombre de usuario que hay en el fichero** por el que quieras usar en las infraestructuras de los laboratorios si lo deseas, aunque no es necesario y puedes dejar simplemente `ssiuser`.



La única carpeta de la que debes preocuparte a partir de ahora es `lab_sessions`

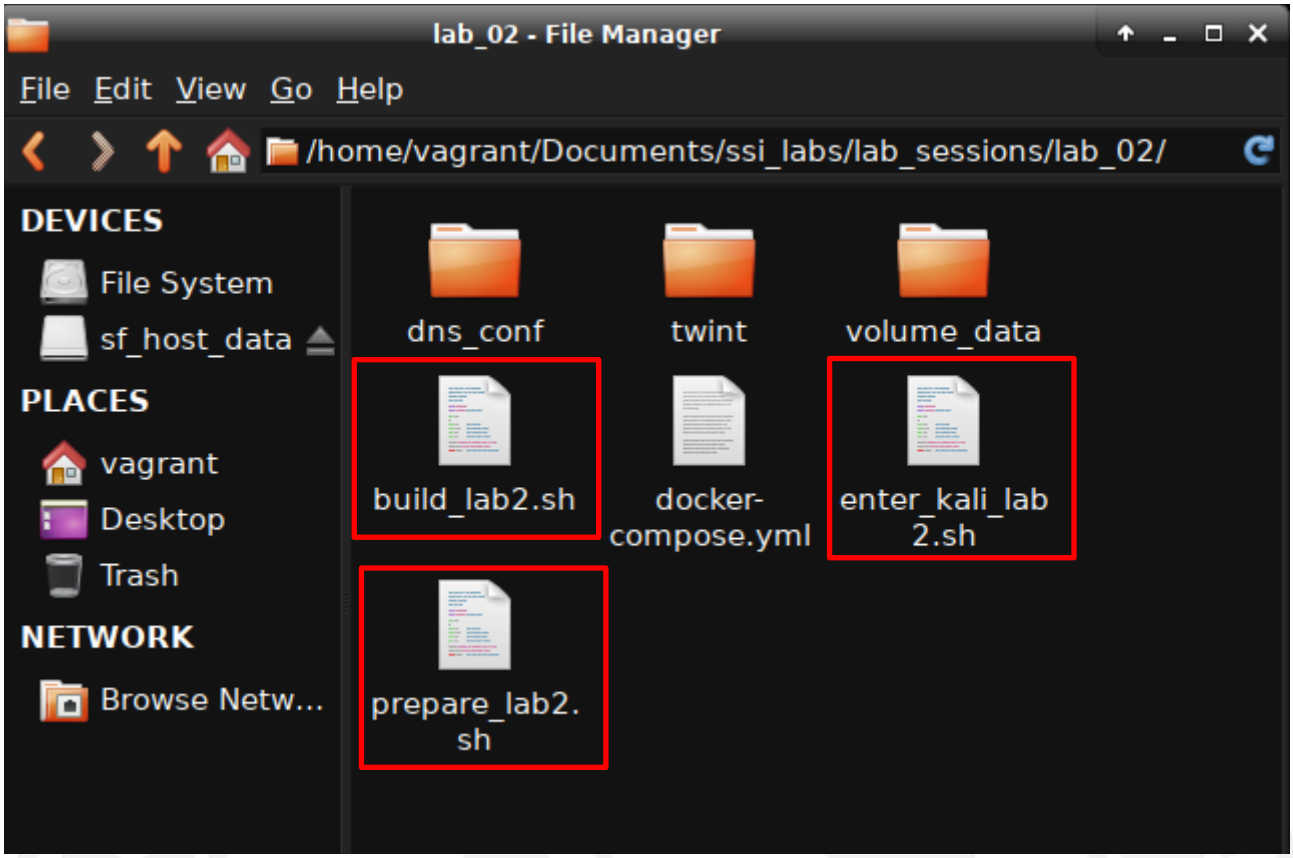


Entra en esta carpeta y en la sesión de laboratorio con la que vas a trabajar. Los números de sesión que faltan significan que esas sesiones utilizan la infraestructura de un laboratorio anterior, o que no requieren una:

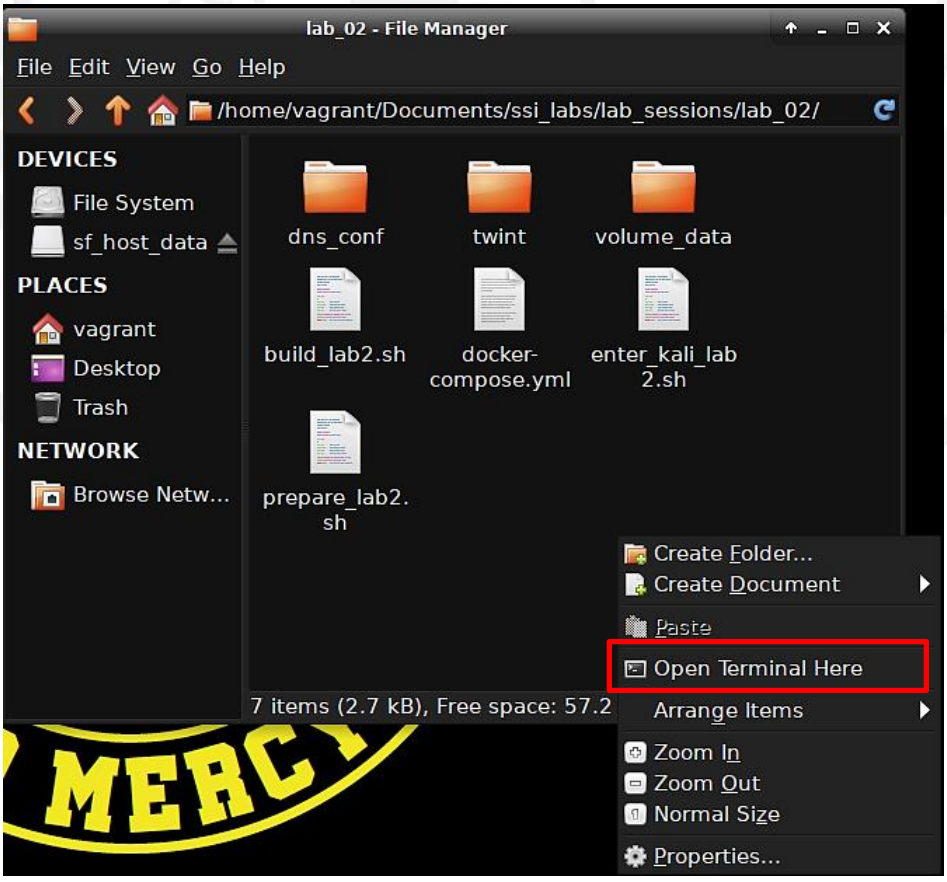


Dentro de las carpetas de laboratorios encontrarás estos elementos, que se detallarán más adelante. El resto de los elementos no presentes en esta lista no son importantes para esta asignatura. **NOTA: ¡todos los laboratorios funcionarán igual que como se explica aquí!**

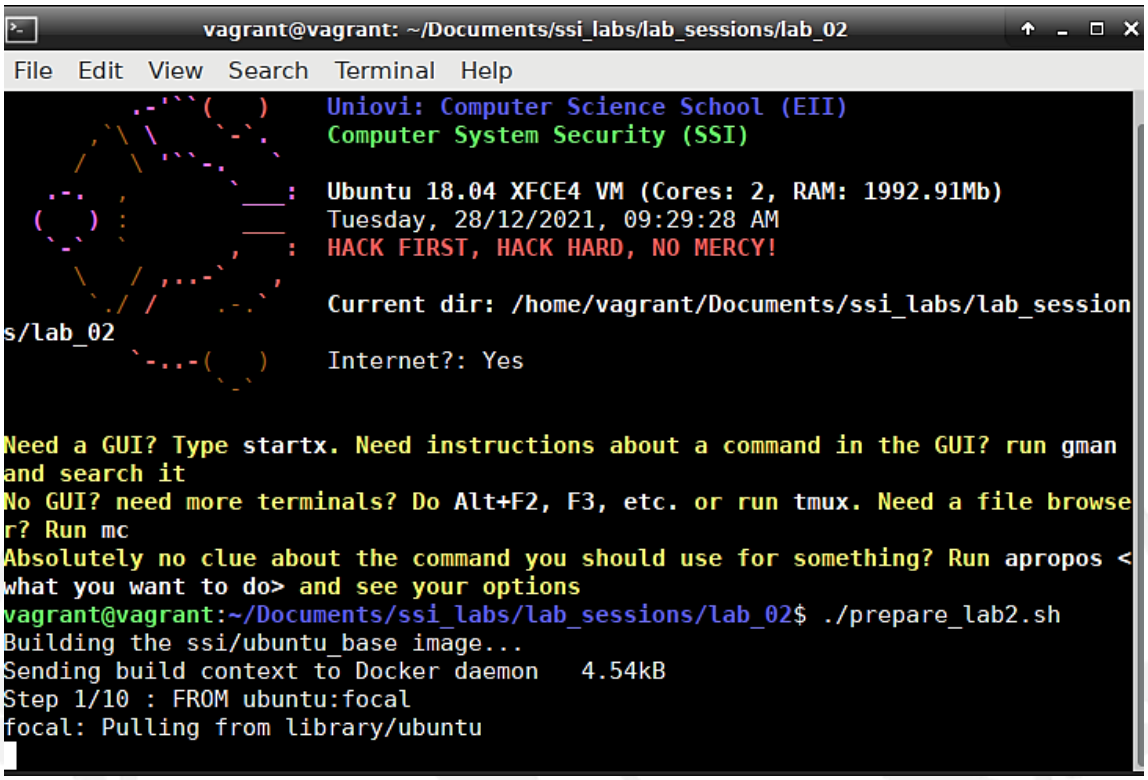
- `prepare_labX.sh`: Este es el archivo que debes **ejecutar primero** para preparar todos los requisitos previos e infraestructuras de laboratorio.
- `build_labX.sh`: Este *script* debe **ejecutarse después de** `prepare_labX.sh` y construye todos los componentes del laboratorio.
- `enter_XXXX.sh`: *Scripts* para **iniciar sesión** en diferentes componentes del laboratorio, según se especifique en las instrucciones de estos. No todos los elementos del laboratorio están pensados para iniciar sesión en ellos, así que no te preocupes si el número de *scripts* de `enter_XXX.sh` es menor que los elementos del laboratorio. **El inicio de sesión usará el mismo nombre de usuario que especificaste antes en** `user.txt`, que también es un usuario *sudoer*. Cada componente de la infraestructura tiene un **nombre de host diferente** que coincide con el diagrama de laboratorio correspondiente para permitirte entender mejor dónde te encuentras. Además, los contenedores *Kali* y *Ubuntu* tienen diferentes "*flavors*" (paleta de colores y *prompt* de *bash*) para distinguirlos visualmente de forma fácil.
- Carpeta `volume_data`: Todo lo que escribas en la carpeta `/shared` de cualquier contenedor en el que puedas iniciar sesión se colocará en la subcarpeta correspondiente de esta (una por cada nombre de *host* de cada contenedor). **En consecuencia, también puedes escribir algo aquí en la máquina virtual y verlo inmediatamente en el contenedor correspondiente.** Esto está pensado para ser un **mecanismo fácil para compartir archivos entre los contenedores y la máquina virtual / host**.



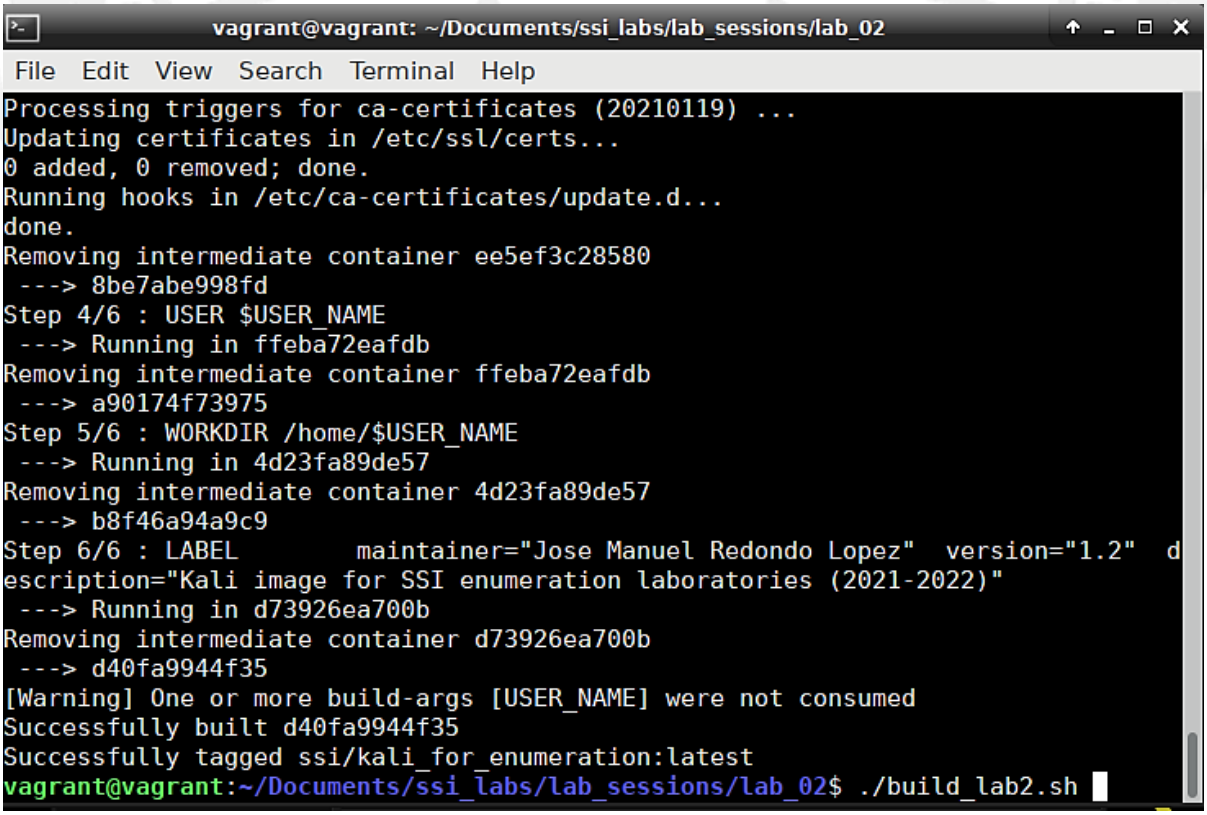
La forma recomendada de trabajar con cada laboratorio es arrancar dos terminales en la carpeta del laboratorio correspondiente de esta manera:



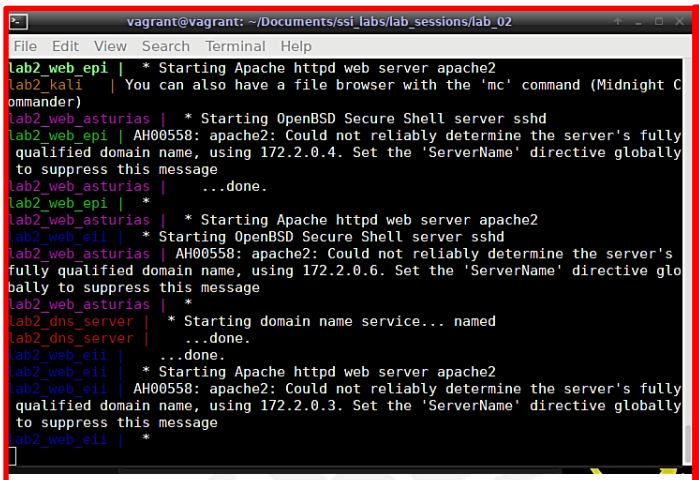
En el primer terminal, ejecuta el *script* `prepare_labX.sh` y espera a que finalice. Solo necesitas hacer esto una vez, pero asegúrate **de tener conexión de red durante el proceso.** Si se pierde, vuelve a ejecutar `prepare_labX.sh`:



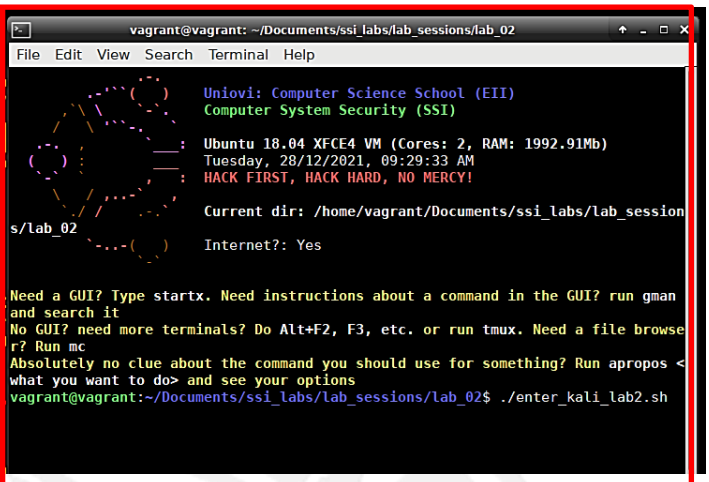
Una vez que esto termine, ejecuta el script `build_labX.sh` y espera a que detenga su salida. **¡No cierres esta terminal!**



Cuando el laboratorio termine de construirse, podrás usar los scripts `enter_XXX.sh` disponibles para iniciar sesión en los contenedores adecuados para el laboratorio actual, utilizando el segundo terminal que abriste:

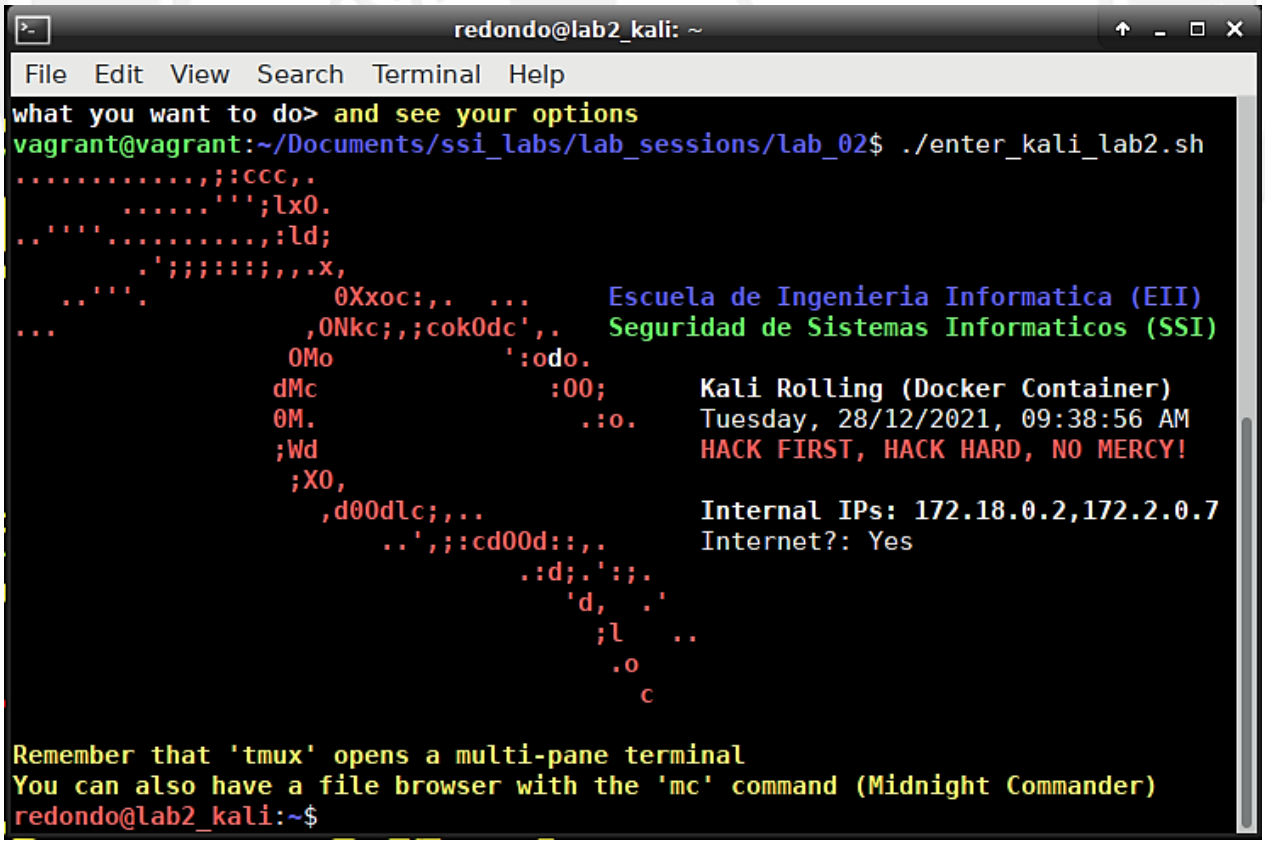


`build_labX.sh`: si cierras esta ventana, se destruirá todo el laboratorio

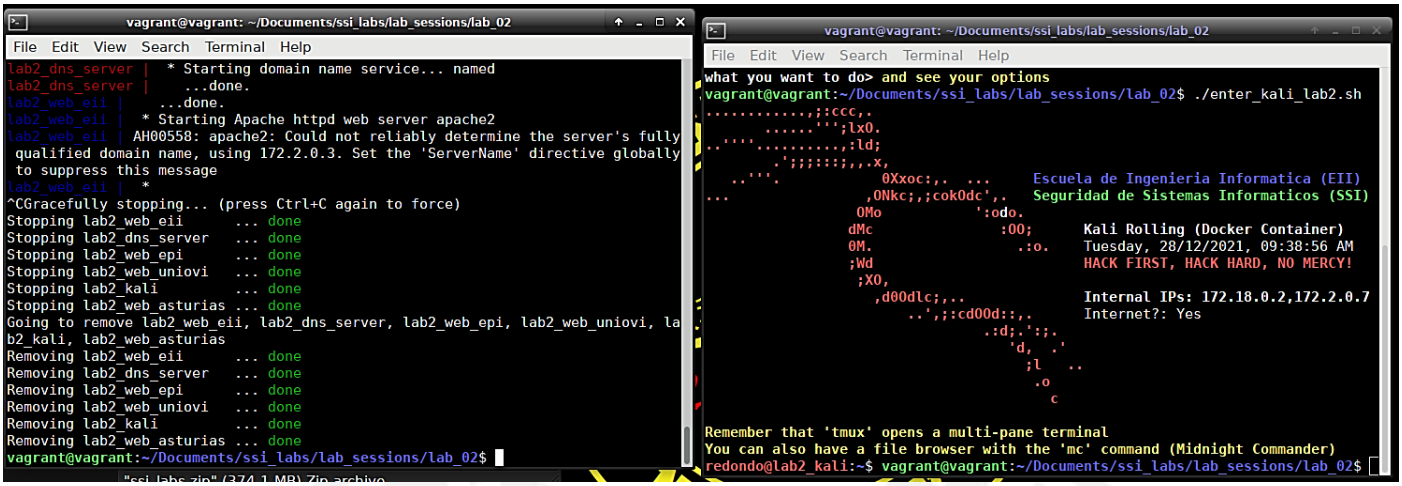


`enter_labX.sh`: si cierras esta venta, el contenedor no se destruirá y puedes entrar de nuevo

¿Observas el cambio de logotipo cuando iniciaste sesión en el contenedor? Ahora eres `<tu nombre de usuario>` (usuario *sudoer*) dentro de un contenedor Kali *Linux* creado para emular una máquina virtual. Sigue las instrucciones de laboratorio correspondientes para comenzar a realizar sus tareas. **¡CADA LABORATORIO TIENE LA MISMA FORMA DE CONFIGURAR SU INFRAESTRUCTURA! ¡A hackear!** 😊



Para parar un laboratorio, vete al terminal donde lo construiste (hiciste el `build_labX.sh`) y **haz Ctrl+C**. Espera a que se complete la finalización (y vuelvas al *bash*, como en la imagen) antes de salir del terminal. Se cerrará automáticamente la sesión de cada contenedor del laboratorio en el que la tengas iniciada actualmente:


vagrant@vagrant: ~/Documents/ssi_labs/lab_sessions/lab_02
File Edit View Search Terminal Help
lab2_dns_server | * Starting domain name service... named
lab2_dns_server | ...done.
lab2_web_eii | ...done.
lab2_web_eii | * Starting Apache httpd web server apache2
lab2_web_eii | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.2.0.3. Set the 'ServerName' directive globally to suppress this message
lab2_web_eii | *
^CGracefully stopping... (press Ctrl+C again to force)
Stopping lab2_web_eii ... done
Stopping lab2_dns_server ... done
Stopping lab2_web_epi ... done
Stopping lab2_web_uniovi ... done
Stopping lab2_kali ... done
Stopping lab2_web_asturias ... done
Going to remove lab2_web_eii, lab2_dns_server, lab2_web_epi, lab2_web_uniovi, lab2_kali, lab2_web_asturias
Removing lab2_web_eii ... done
Removing lab2_dns_server ... done
Removing lab2_web_epi ... done
Removing lab2_web_uniovi ... done
Removing lab2_kali ... done
Removing lab2_web_asturias ... done
vagrant@vagrant:~/Documents/ssi_labs/lab_sessions/lab_02$
vagrant@vagrant: ~/Documents/ssi_labs/lab_sessions/lab_02
File Edit View Search Terminal Help
what you want to do> and see your options
vagrant@vagrant:~/Documents/ssi_labs/lab_sessions/lab_02$ ./enter_kali_lab2.sh
Escuela de Ingenieria Informatica (EII)
Seguridad de Sistemas Informaticos (SSI)
Kali Rolling (Docker Container)
Tuesday, 28/12/2021, 09:38:56 AM
HACK FIRST, HACK HARD, NO MERCY!
Internal IPs: 172.18.0.2,172.2.0.7
Internet?: Yes
Remember that 'tmux' opens a multi-pane terminal
You can also have a file browser with the 'mc' command (Midnight Commander)
redondo@lab2_kali:~$ vagrant@vagrant:~/Documents/ssi_labs/lab_sessions/lab_02$