Sharing Files Between the Host OS and a Linux VM

This tutorial uses Parrot Security Edition but the instructions are the same for Kali Linux

---

When beginning your journey with the Cyber Security content on the INE Platform, students are told numerous times that Linux is the way to go. We feel it is vastly important to instill this early and often, as this is what being successful in the field demands. With many things in life, there are exceptions to every rule. In this case, as students progress past the first recommended Learning Path of Penetration Testing Student (PTS), there may be an occasion to use their local OS, Windows or OSX. Even though you've heard us go on incessantly about hacker tools and their penchant for being on Linux, there are admittedly some commercial tools (and their free, community editions) that are not available for Linux at all. Students may also find that they prefer using their host OS to navigate my.ine.com, study the course materials and control the labs, then move to Linux to complete the goals of the labs. Either way, there rises the need to move files between the Linux VM and your host OS.

On the bright side, using our recommendation of importing a virtual appliance version of either Parrot or Kali in VirtualBox with the VM VirtualBox Extension Pack, sharing files can be done in two ways:

1. Simply copy and paste files between the VM and the host
2. Setup a shared folder (sf) between the VM and the host

Since option #1 works immediately as is, there's no further setup needed. However, Step #2 of using a shared folder and then automating the connection every time one starts their VM takes a little

setup. It's not complicated but has several steps that may confuse newcomers. This tutorial shows how to accomplish this.

Prerequisites for This Tutorial

- VirtualBox with Extension Pack installed on your local OS
- A dedicated folder/directory on your local OS for sharing
- Virtual Appliance version of Parrot or Kali VM running in VirtualBox

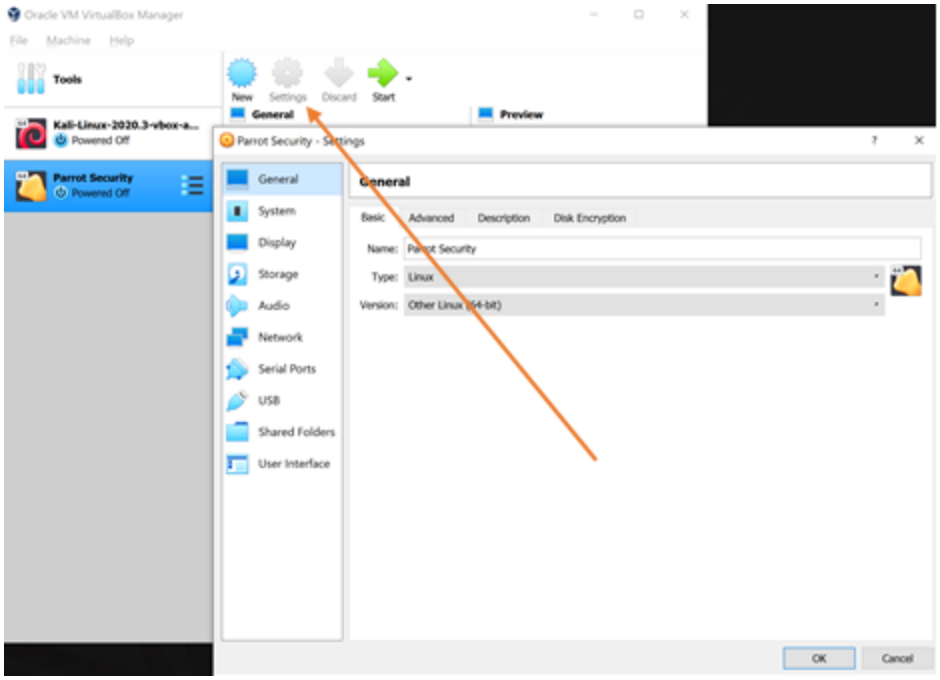
Quick Steps

1. Open VirtualBox and go to the "Settings" for your VM
2. On the "Shared Folders" tab, Add a new shared folder using the location of your predetermined local folder. Be sure to select the "Auto-mount" option.
3. Launch your VM and verify that you see a shortcut on the desktop for "sf_[folder_name]".
4. Add permission to the shared folder by opening a terminal window and using the following command: "sudo adduser [username] vboxsf"
5. Logoff and log back in
6. Open "sf_[folder_name]" and you should have access with no permission errors

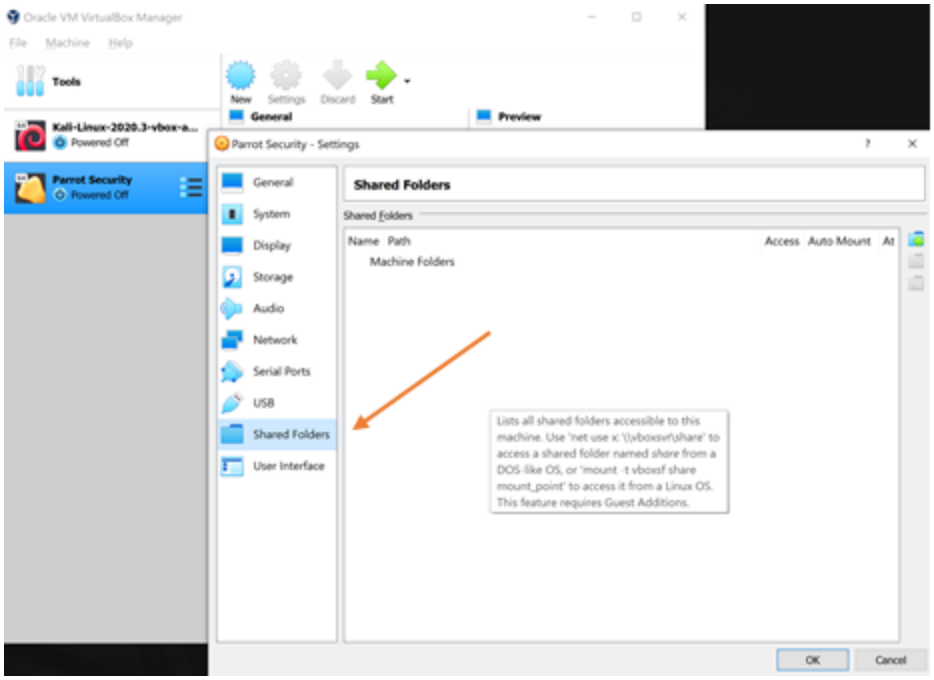
Detailed Steps

The following step-by-step instructions are exactly the same as the Quick Steps but are accompanied with a lot more detail. This tutorial is done with a Parrot Security VM; however, the steps in your host OS, VirtualBox and the Linux commands are the same when using Kali except username and password.

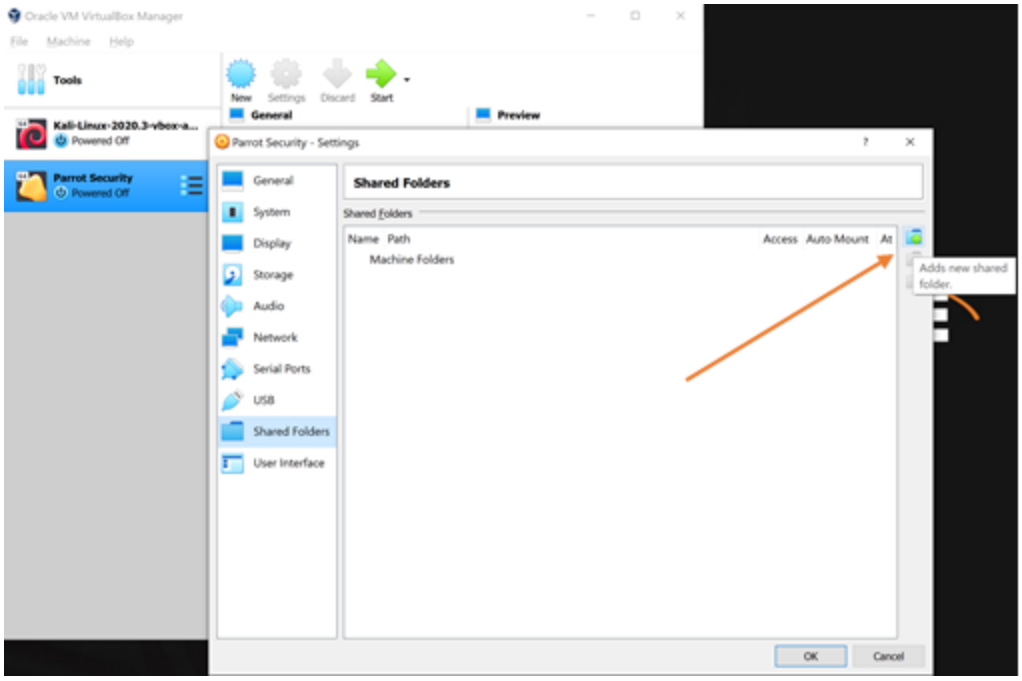
1. Open VirtualBox and go to the "Settings" for your VM



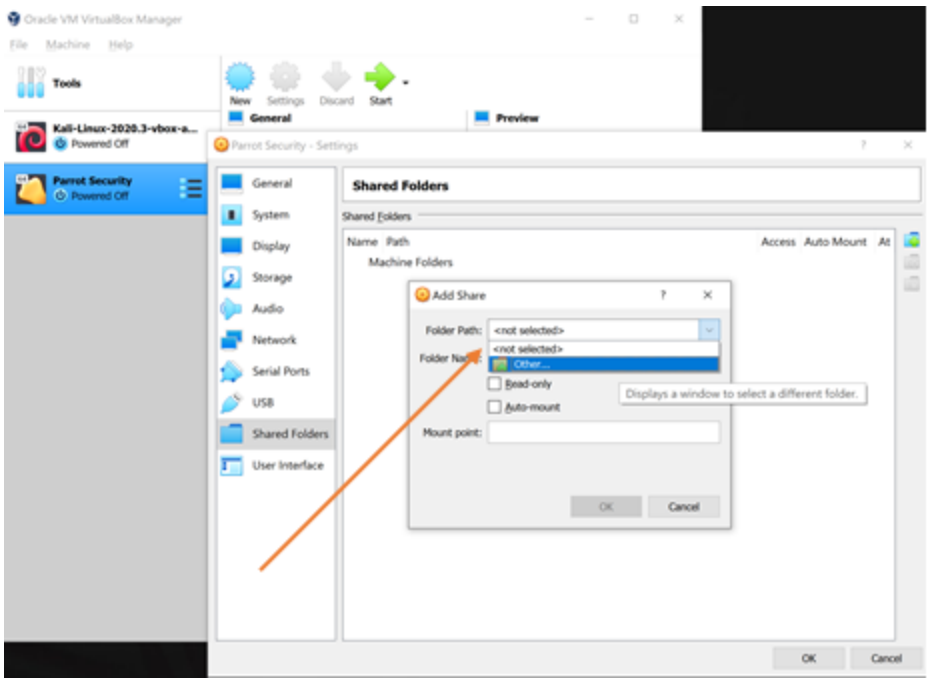
2. Open the “Shared Folders” tab.



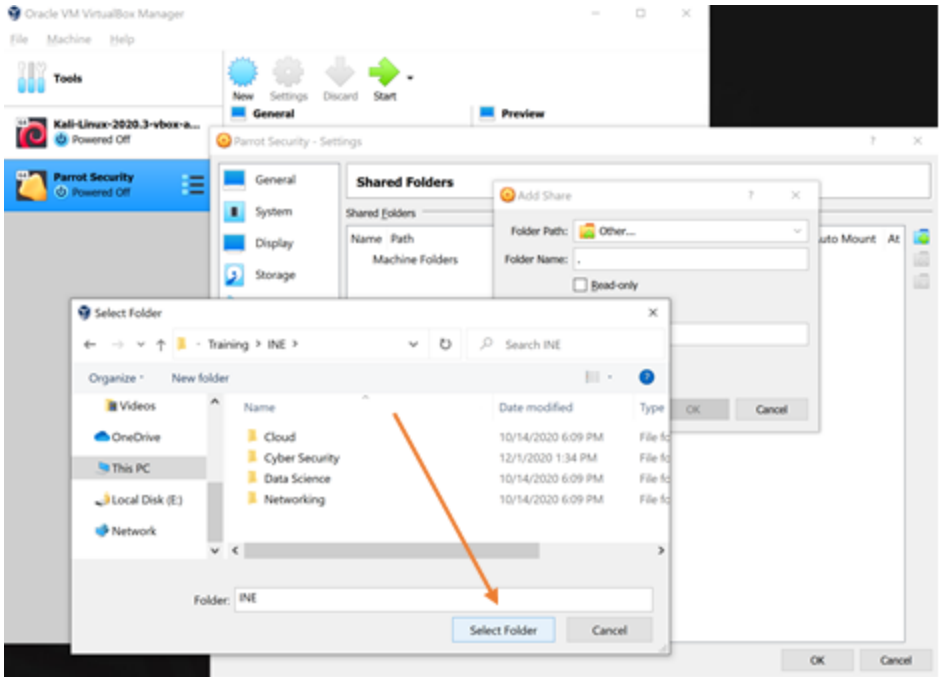
Add a new shared folder by clicking the icon on the right with the blue folder and the green plus sign.



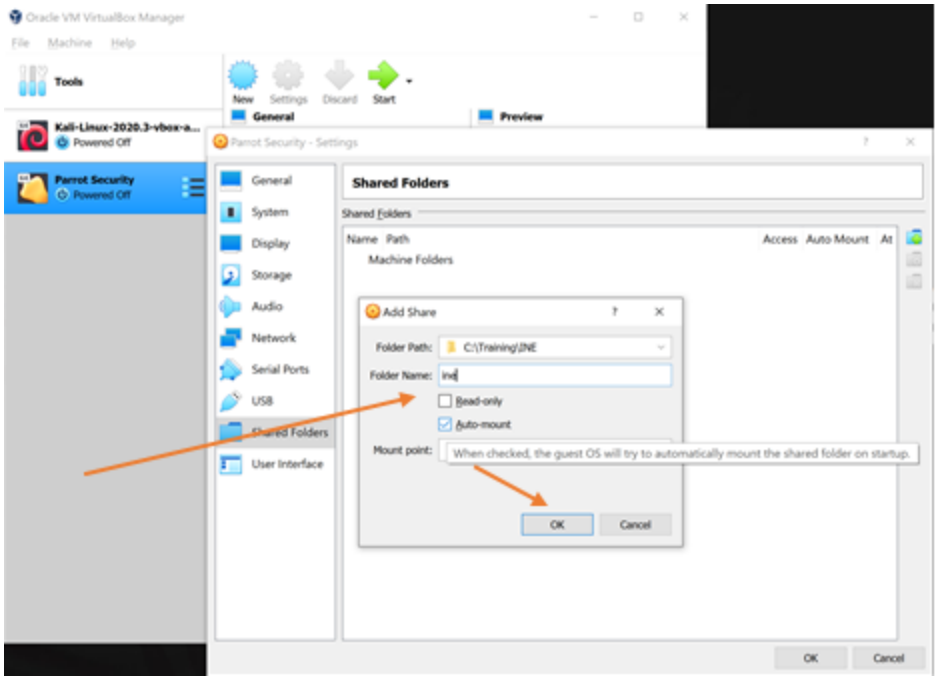
Under “Folder Path”, click the down arrow and select “Other”.



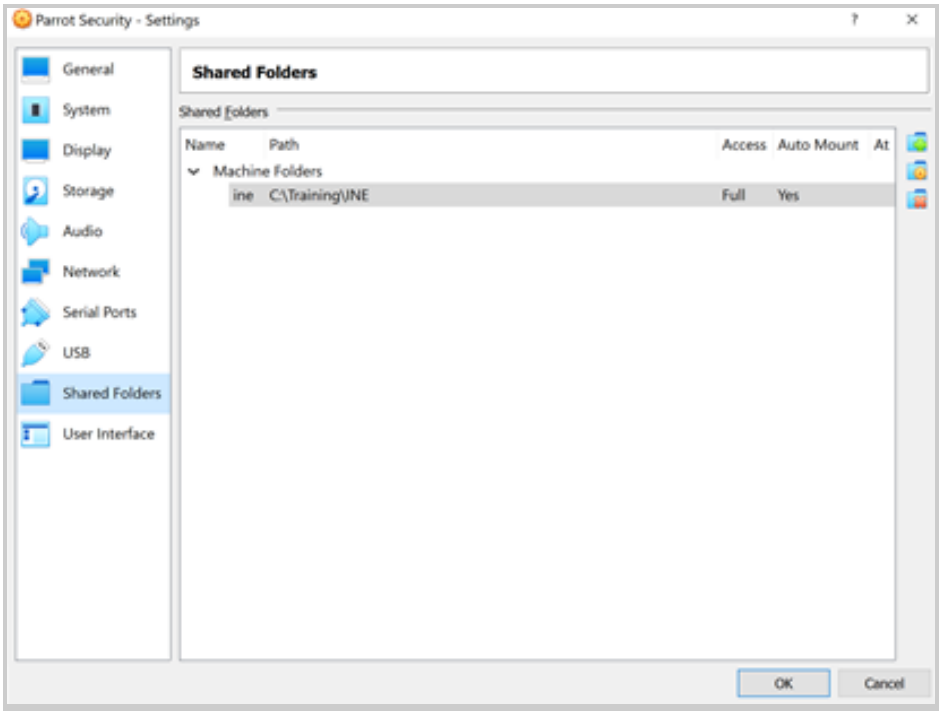
Navigate to the location of your predetermined local folder and click “Select Folder”.



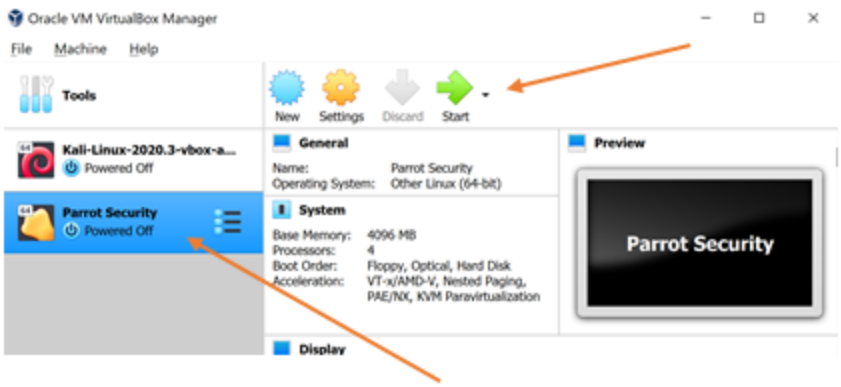
Notice it adds the path to your local folder but also assigns a name that matches the folder name. In this case, the Windows folder is “INE”. Since Linux is case sensitive and is easier to type in lower-case from the Linux command line, it is recommended to make this lower case, “ine”. A debate on making your local drive “Read-only” is for another time. For now, leave this unchecked. Be sure to select the “Auto-mount” option in order for Parrot to automatically connect to the shared folder. Click “OK”.



You have successfully set up a shared folder. Remember that this is abbreviated as “sf”.

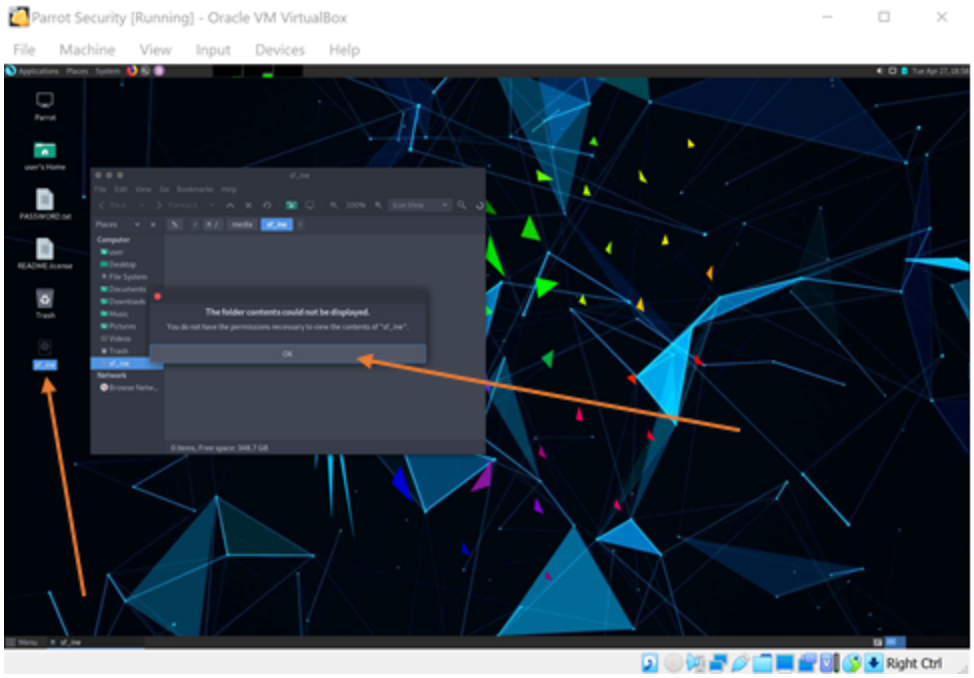


3. “Start” your VM by highlighting Parrot Security on the left-hand side and then clicking the large green arrow.

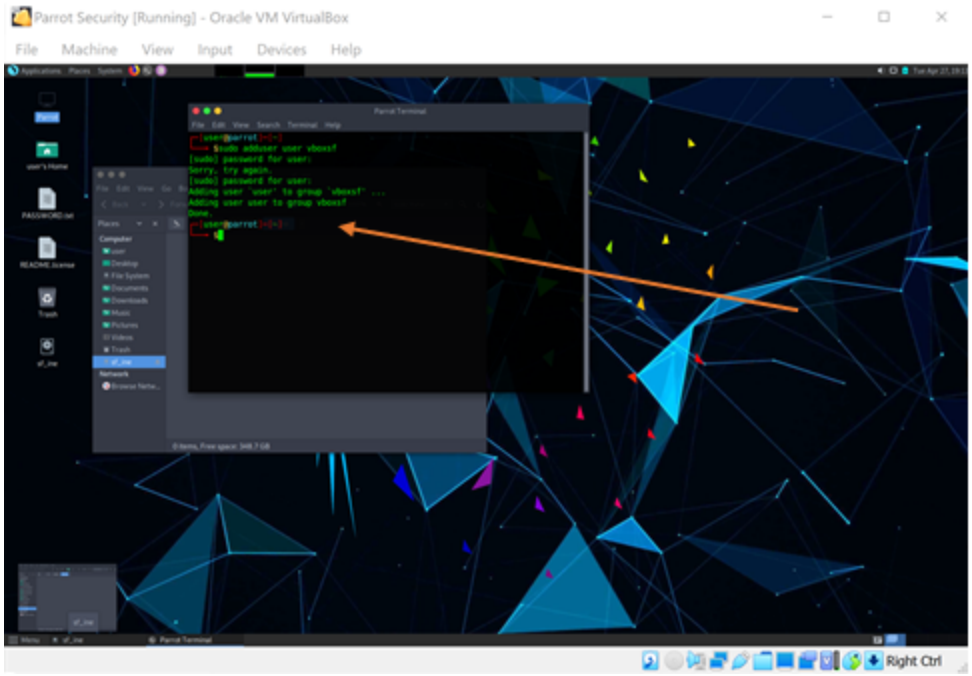


After the normal bootup sequence, we are presented with the login screen briefly and will then be taken directly to the desktop with no interaction. Because it is the virtual appliance, a password is not required to login. Changing this behavior would be a good idea, but we’ll leave it this way for now. Looking at the desktop, we can verify that you see a shortcut on the desktop for “sf_[folder_name]”. In our case it is “sf_ine” all lower-case.

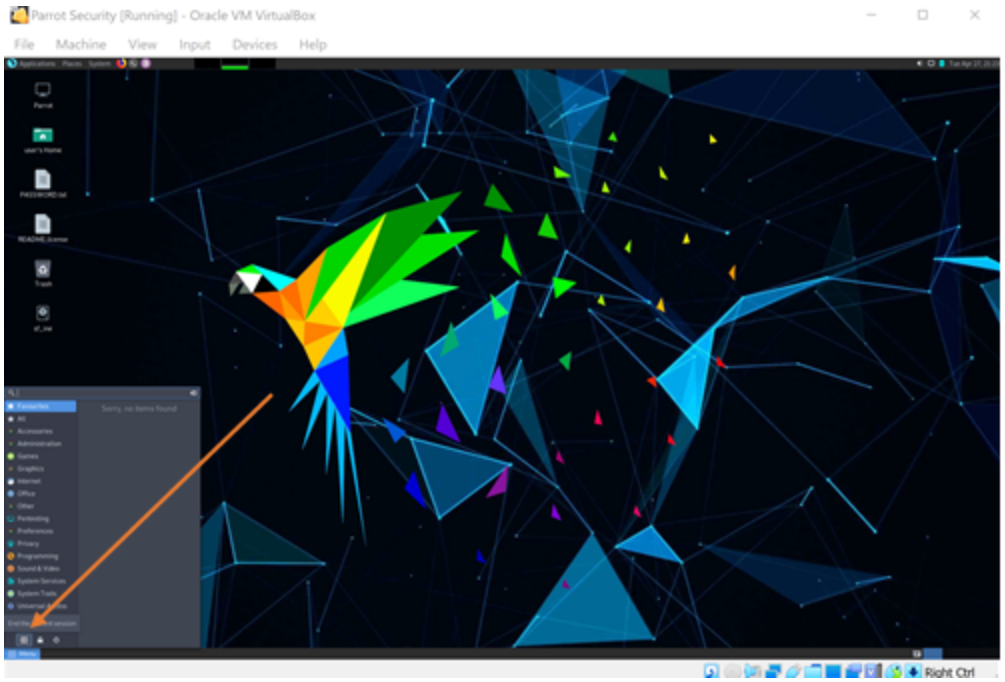
However, when double-clicking the folder to open it, we are denied access. Don’t worry. This is normal. We need to add our user to the VirtualBox Shared Folder group which is aptly named “vboxsf”.



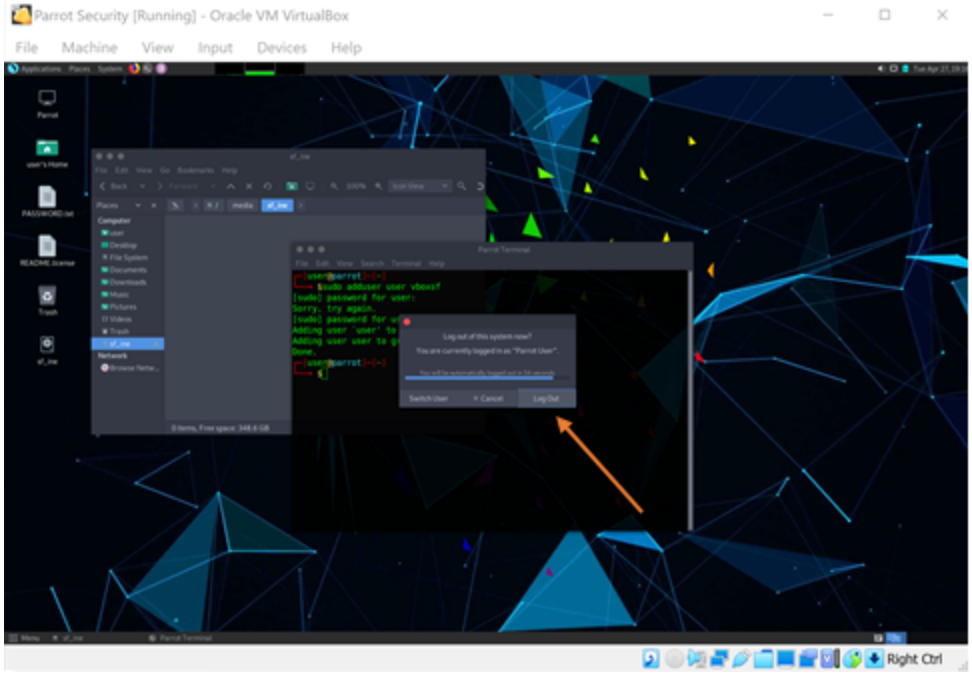
4. To add permission to the shared folder group, let's open a terminal window. In the standard MATE Edition of Parrot, the terminal icon with ">_" can be found at the top menu next to the Firefox icon. Open it and use the following command: "sudo adduser [username] vboxsf". The default limited user in Parrot is named "user"; therefore, in our case, we would type: "sudo adduser user vboxsf". Since the "sudo" command requires elevated privileges, we are asked for a password for the "user" account. According to the Parrot documentation, the default password for the "user" account is "live". That did not work in our tests. The documentation continues that if "live" does not work, try "toor". That does the trick, the password is accepted, and our command runs successfully.



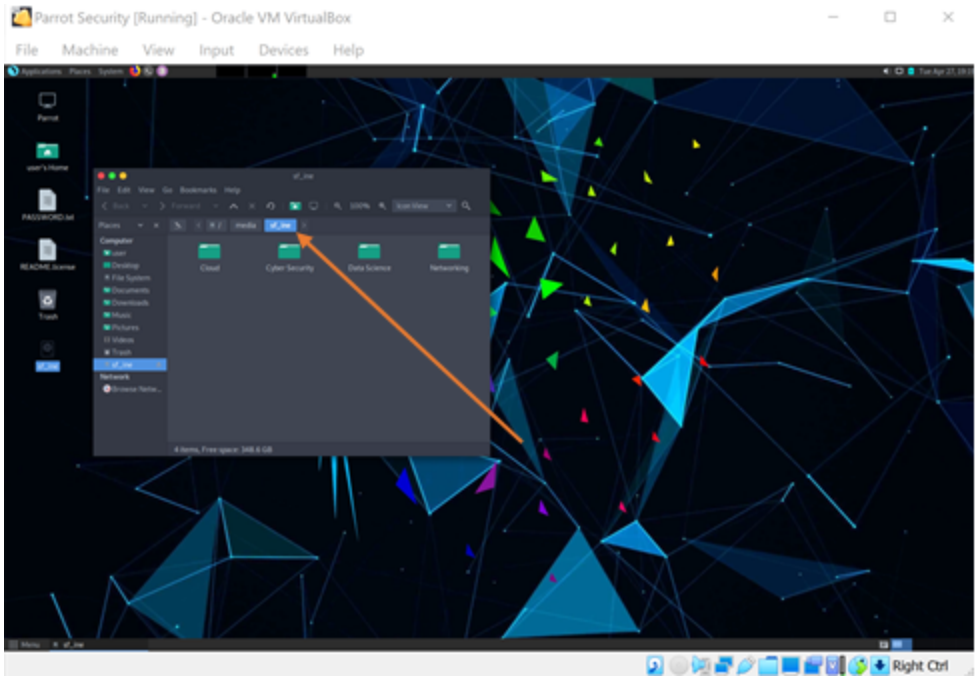
5. Because this is a change to the account settings, we will not realize our new group membership until we logoff and log back in. A restart of the VM is not required. In Parrot, click the “Menu” icon at the far bottom left of the desktop and click the far left icon that looks like a box with an arrow to “End the current session”.



Click “Log Out” to complete the process.



6. You will then be taken back to the original login screen, which will again log you in automatically to the desktop. Double-clicking on “sf_ine” is now successful, and you should have full access with no permission errors



Now you have an easy way to copy files between your VM and your host machine. Using the examples from above, you may now:

1. Grab files from a lab machine from Parrot or Kali and analyze using tools on your host OS.
2. Study the materials from your host OS as well as start labs. Since starting the labs automatically generates an ovpn file, you could copy the ovpn file to the shared folder on your host machine and have direct access to it from Linux to complete the lab goals.
3. Hop between the 2 as needed based on the tasks at hand.

Happy Hacking!

Content Team

.