LAB DESCRIPTION

SETTING UP A VULNERABLE SCADA/ICS LAB

By TheWardonianEffect

Last Update: 30 May, 2020

LAB SETUP

REQUIRED DOWNLOADS

Note: this lab was successfully conducted using a Windows 10 host OS. Please download the software pertaining to your particular machine's OS. It is recommended that you download all of the files into a folder you can easily access as we begin to import the VMs into VirtualBox.

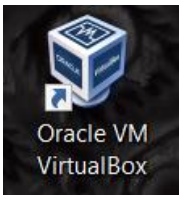
To run this lab, you will need to setup your environment by downloading the software outlined in the links below. Be aware that some links may be broken depending on how often a company updates their website, so if all else fails then Google will be your best bet.

- VirtualBox Platform Package https://www.virtualbox.org/wiki/Downloads
- VirtualBox Extension Pack https://www.virtualbox.org/wiki/Downloads
- ScadaBR VM (the SCADA/ICS server) https://openplcproject.com/reference/scadabr/
- Windows 10 VM https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/
- Kali Linux 64-bit VirtualBox VM https://www.kali.org/downloads/

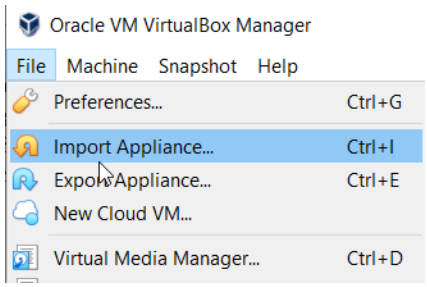
SCADABR SETUP

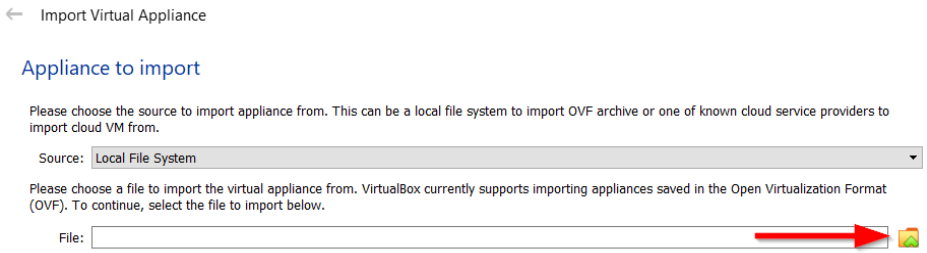
The ScadaBR VM will be acting as the SCADA/ICS server administered by the Windows 10 VM. Once running, users with the right credentials will be able to log into the ScadaBR website to add, remove, or modify SCADA/ICS data sources. We will also enable SSH in this VM to simulate remote administration by the Windows 10 VM, and also to enable different attacks possible by our Kali VM.

1) Open VirtualBox

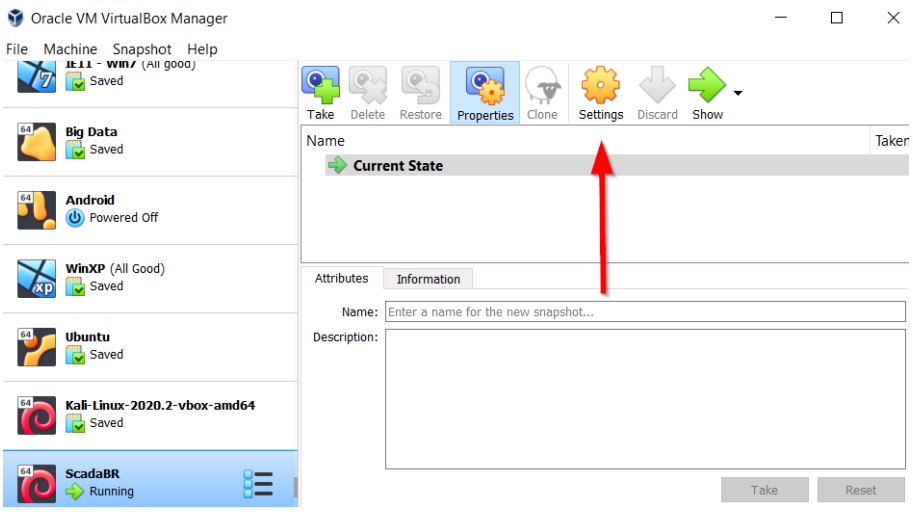


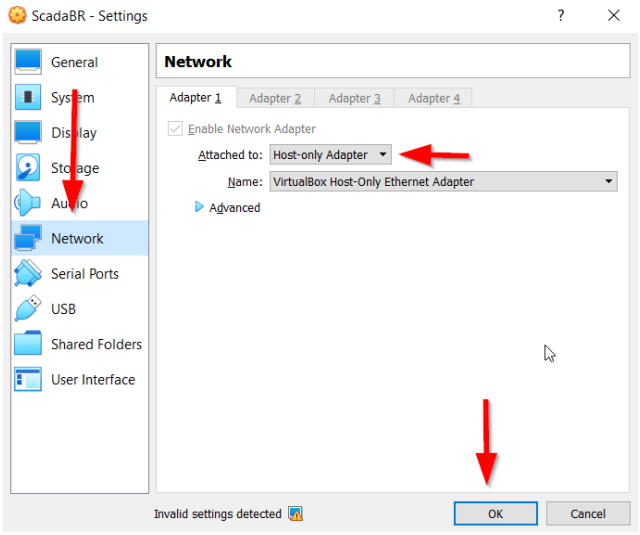
2) With VirtualBox open, select **File > Import Appliance**, and click on the **yellow folder with the green arrow** to import the ScadaBR VM.



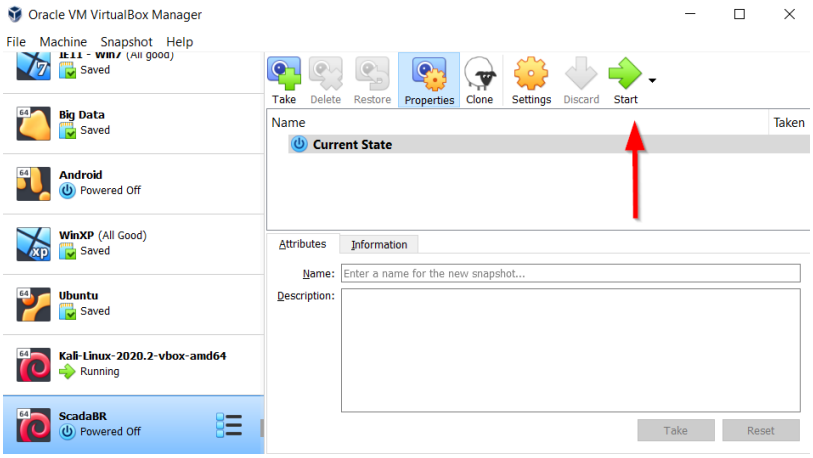


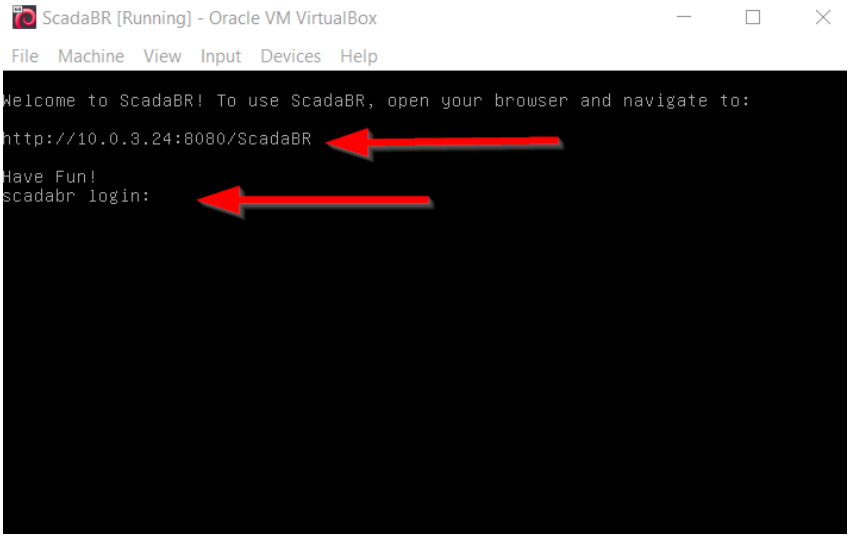
3) Select your ScadaBR ova file, click **Next**, and finally click **Import** on the Appliance Settings screen.
4) Once the ova file is imported, select the ScadaBR VM, click on **Settings** (the yellow cog) > **Network** > select **Attached to: Host-Only Adapter** in the drop down menu > hit **Ok**



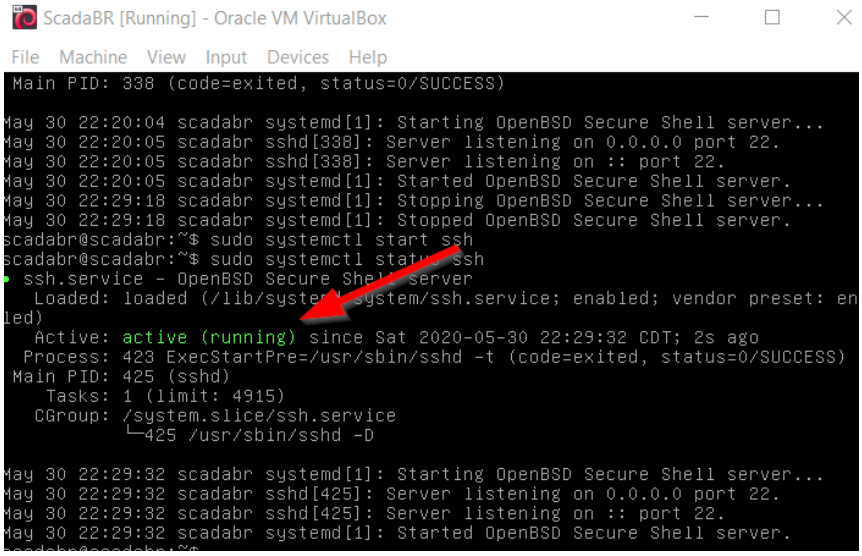


5) Once the network settings are configured, go back to the main VirtualBox menu, highlight the ScadaBR VM, and start it by clicking on the **Start** icon (green arrow). After a couple of seconds, you should be prompted with the ScadaBR's administrative link along with the local administration login prompt. Note: your ScadaBR VM's IP may be different from what you see on this document.



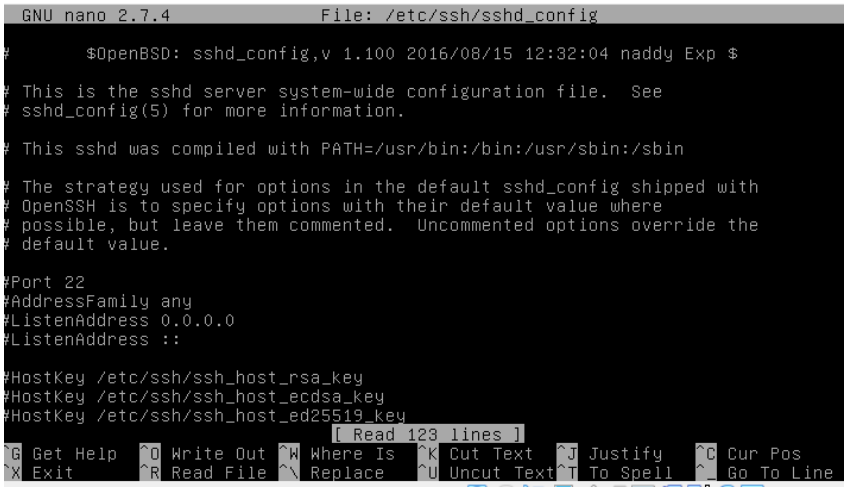


6) To enable SSH, login with the credentials **root** (username), and **scadabr** (password). Note: you will not see the password characters being typed.
7) Once logged in, you will need to install SSH. Type the following command in the terminal and hit Enter. If prompted for a password, just type **scadabr** and hit Enter.
    - *apt install openssh-server*
8) To ensure that SSH is running, type the following in the terminal and hit Enter. You should see active (running) text in the terminal.
    - *systemctl status ssh*

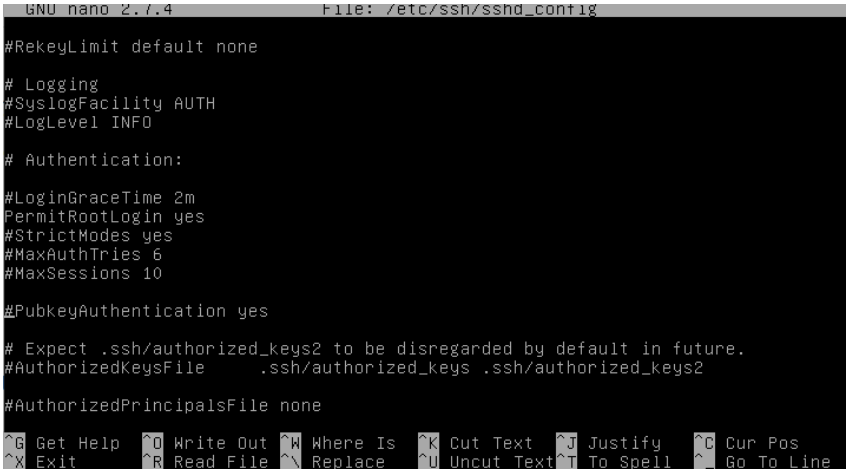


9) If for whatever reason SSH is not running, just enter the following in the terminal
    - *sudo systemctl start ssh*
10) We will now enable root login over SSH so our Windows 10 VM can log into the ScadaBR server to make any administrative changes. To do so, you will need to edit the ssh configuration file. Type the following in the terminal:
    a. *nano /etc/ssh/sshd_config*

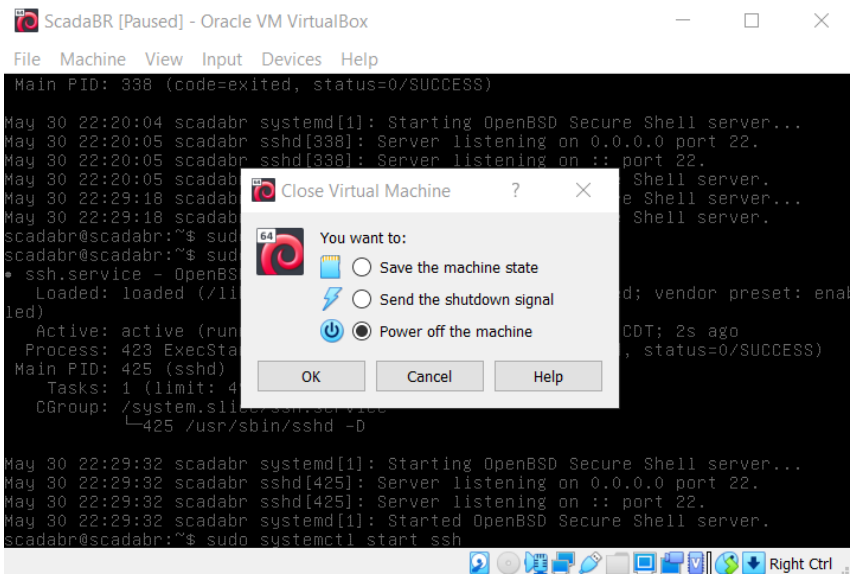
11) At this point, you should see something like what you see in the image below



12) Using the **down arrow**, go all the way where it says *PermitRootLogin*. Remove the # sign and ensure that all it says after *PermitRootLogin* is *yes*. It should look like the image below



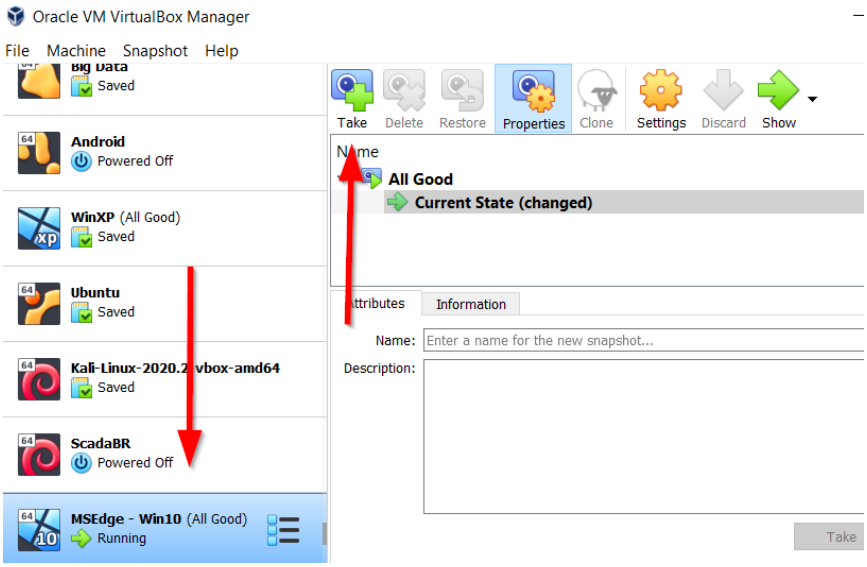
13) Save the updated configuration file by holding **CTRL** and typing the letter **X**. When prompted whether or not you want to save the changed file, type **yes**, hit Enter, and then hit Enter once again to keep the file name. Do not change the file name!
14) Exit out of the ScadaBR VM and select **Power off the machine**.



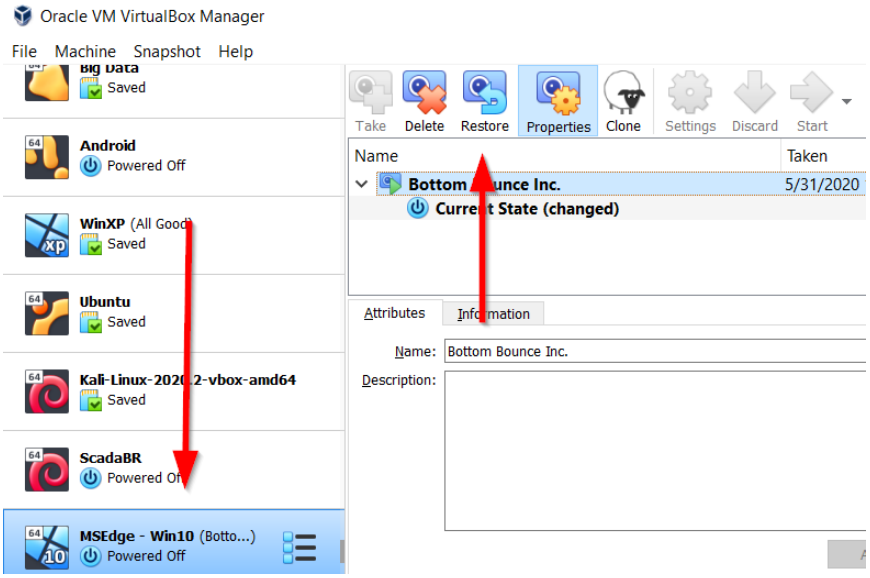
WINDOWS 10 SETUP

The Windows 10 VM will be the machine responsible for administering the SCADA/ICS data sources (i.e. temperature reader, on/off switches, etc) and remotely administering the ScadaBR server via SSH. The import method all the way to setting up the network is the same as the ScadaBR's, with the exception that we will be selecting the Windows 10 ova file instead. Please import the file and setup the network settings before proceeding with the next steps.

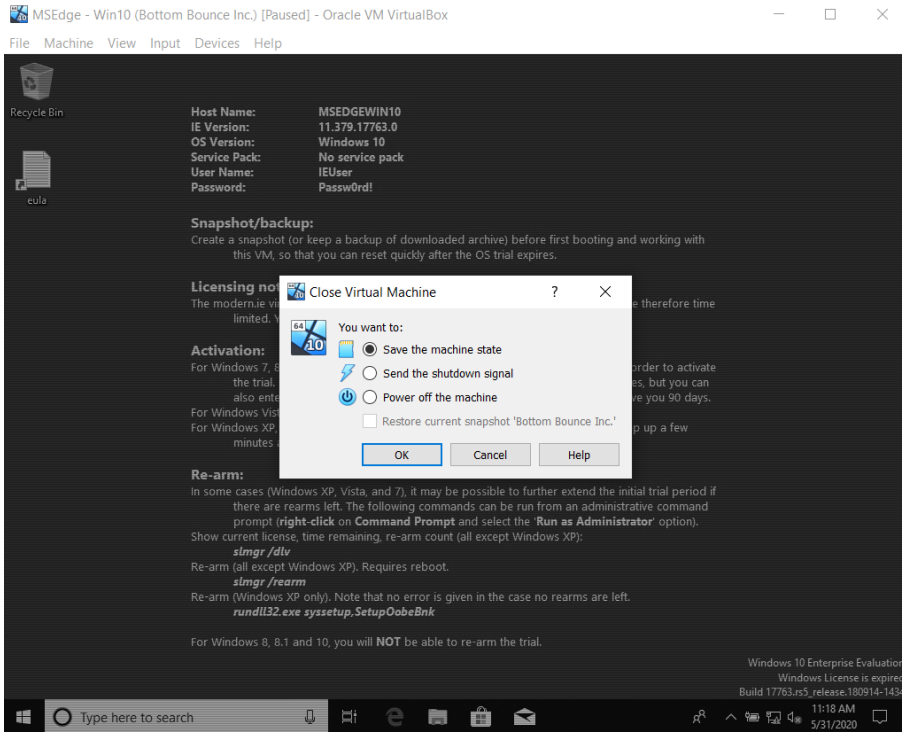
1) After following the same importing and network setting steps found for the ScadaBR, click on the **Start** icon (green arrow) to launch the VM. At the login page, enter the password **Password!**
2) Note: because this is an evaluation VM, you only have 90 days to use it to its full extent before the need to activate it. For this reason, create a snapshot by doing the following:
    - Go to the main VirtualBox dashboard, highlight the Windows 10 VM, and select **Take** (the camera with the green + sign in front)



- For the purpose of these labs, we will name the snapshot *Bottom Bounce Inc.*
- If at any point your VM becomes corrupted for some reason, or the trial expires, simply turn the VM off, highlight the VM on the VirtualBox dashboard, and then click on the **Restore** icon (camera with blue arrow) to restore the image to how you last saved it.



3) Now that you are done setting up the Windows 10 VM, exit the VM, except that this time you will select **Save the machine state** so you can pick up where you left off (without logging back in) next time you launch the VM.



KALI LINUX SETUP

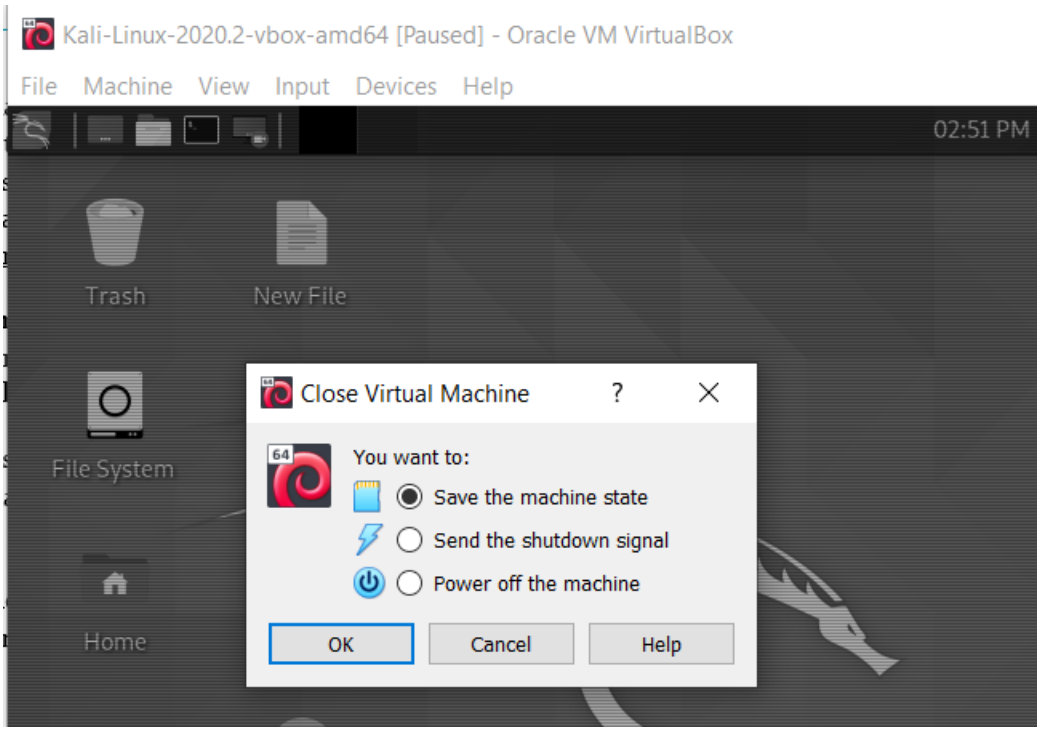
The Kali Linux VM will be your attacking machine. It contains a myriad of tools commonly used in penetration testing, red teaming, etc, but we will focus only on a few tools that will be covered later in this guide. The importing process and network setup is the same as the ScadaBR and the Windows 10 VMs, with the exception that we will be importing the Kali Linux ova file instead. Please import the file and setup the network settings before proceeding with the next steps.

Note: do not use any these tools outside of your lab environment! Any action you take upon the information on this lab is strictly at your own risk, and we will not be liable for any losses and/or damages in connection with this lab.

1) After following the same importing and network setting steps found for the ScadaBR, click on the **Start** icon (green arrow) to launch the VM. The username is **kali** and the password is **kali**.

2) On the Desktop, **right click** anywhere and select **Create Folder**. For the purposes of this lab, name the folder *bottombounce*. You can name it however you'd like, as long as you know where the material for these labs will be located.

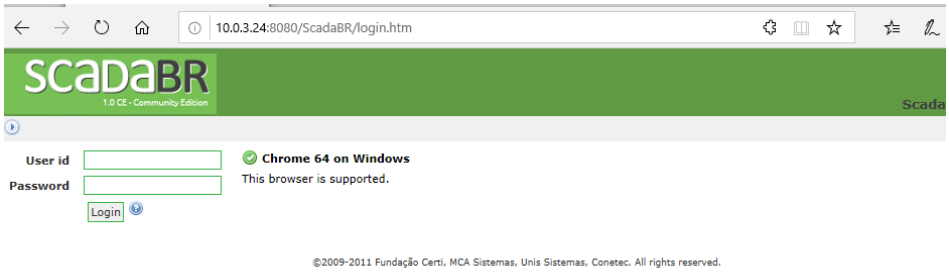
3) Exit the VM and select **Save the machine state** so you can pick up where you left off (without logging back in) next time you launch the VM.



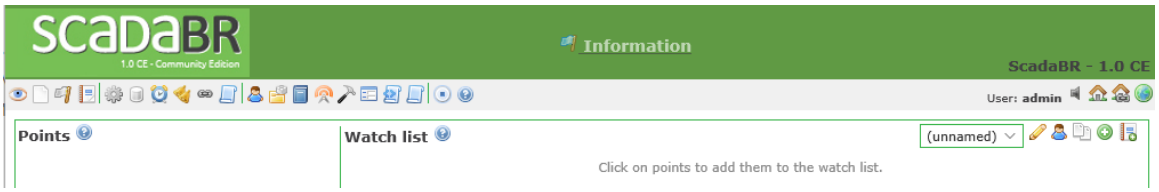
CREATING A DATA SOURCE

Data source, as the name implies, is what will be generating the data (i.e. temperature, on/off, etc) that our attacking machine will go after. More information about data sources and everything else pertaining to ScadaBR can be found at https://sourceforge.net/p/scadabr/wiki/Manual%20ScadaBR%20English%200%20Summary/

1) Launch the ScadaBR VM and take note of the IP address shown at login. It should follow the format of *http://<IP>:8080/ScadaBR*. Note: your VMs IP address may be different from what is shown here.
2) Launch the Windows 10 VM, and navigate the ScadaBR admin page through Internet Explorer. You should be welcomed by the ScadaBR admin page as seen below

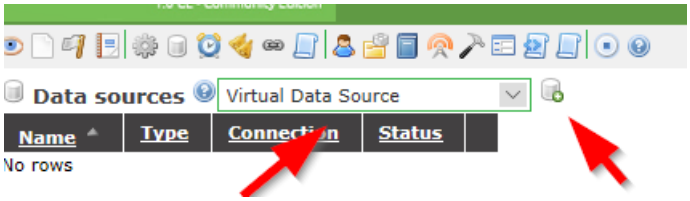


3) Login with User id **admin** and Password **admin**. You should be welcomed with the ScadaBR dashboard as shown below. We will not go into the different details of what each icon means, but you can always check the ScadaBR manual link provided in the beginning of this section.

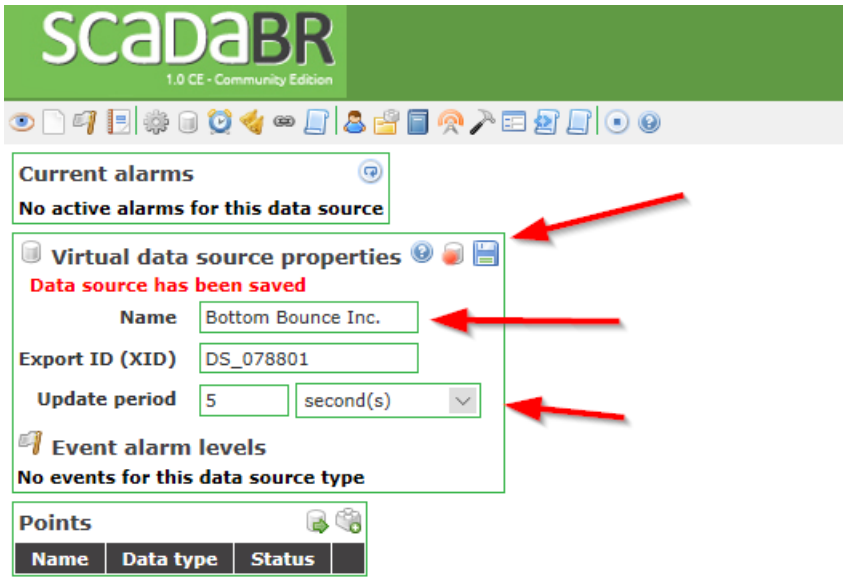


4) While on the dashboard, click on **Data Sources.** As the name implies, this is the main page where all of the administrator's data sources will be created, modified, or deleted.

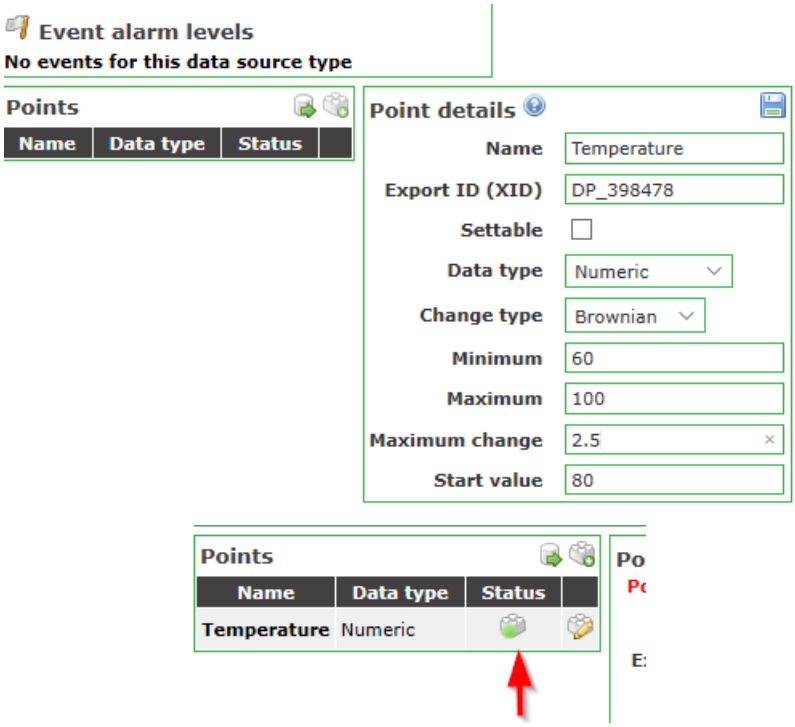
5) On the drop down menu, select **Virtual Data Source** and then click on the **Add** icon next to it.



6) For the **Virtual data source properties**, name the data source as **Bottom Bounce Inc.**, select the **Update period** as **second(s)**, and then click on **Save.** As you might expect, we just created a data source that will give us updates every five seconds.

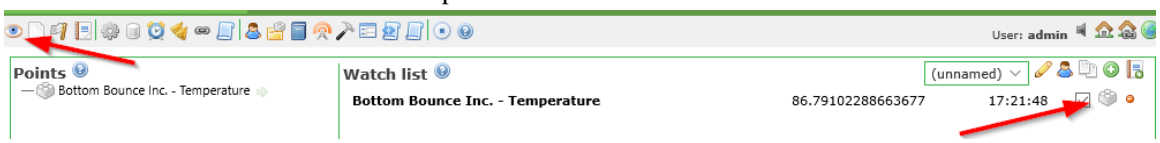


7) Now we need to edit exactly what kind of values will be given every five seconds by the data source. To do this, click on **Point details**, and input the values as you see below. What we did is create a temperature device whose temperature readings ranges from 60 to 100 units, with a maximum change of 2.5 degrees and starting at 80 degrees (median). Once done, click on **Save** (the blue floppy disk) and then enable the data source.



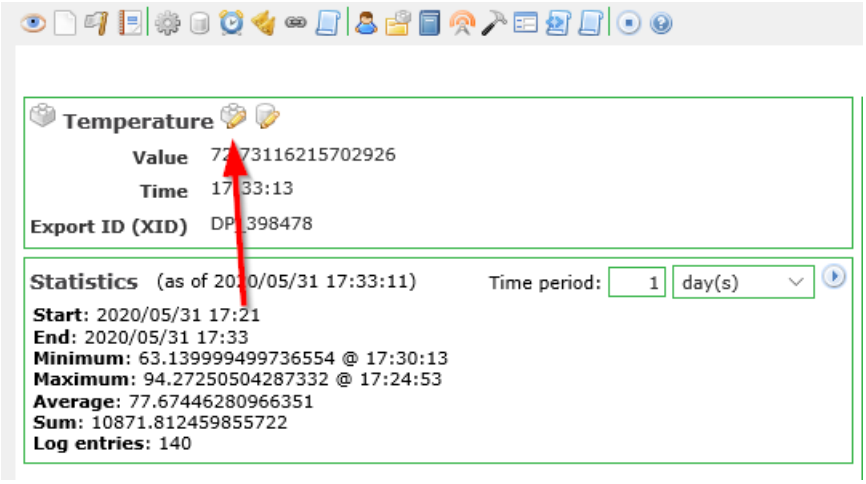
8) Now that your data source is enabled, click on **Watch list** (eye symbol) on the ScadaBR dashboard. You should see the Bottom Bounce Inc. temperature data source on your screen, as well as the readings on the Watch list screen. It all the readings are present, click

on **Point details**. Note: if no readings are present, go back to the **Data sources** tab and ensure the Bottom Bounce Inc. – Temperature data source is enabled.

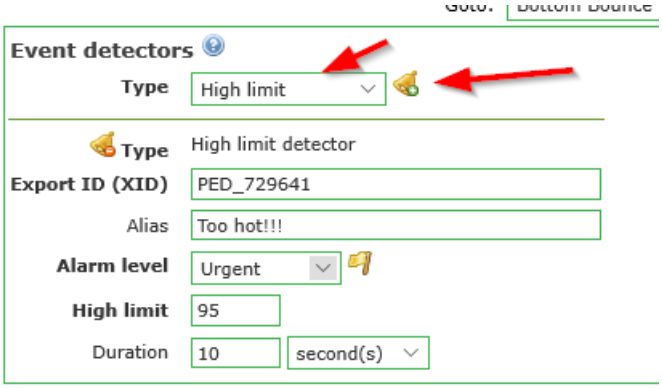


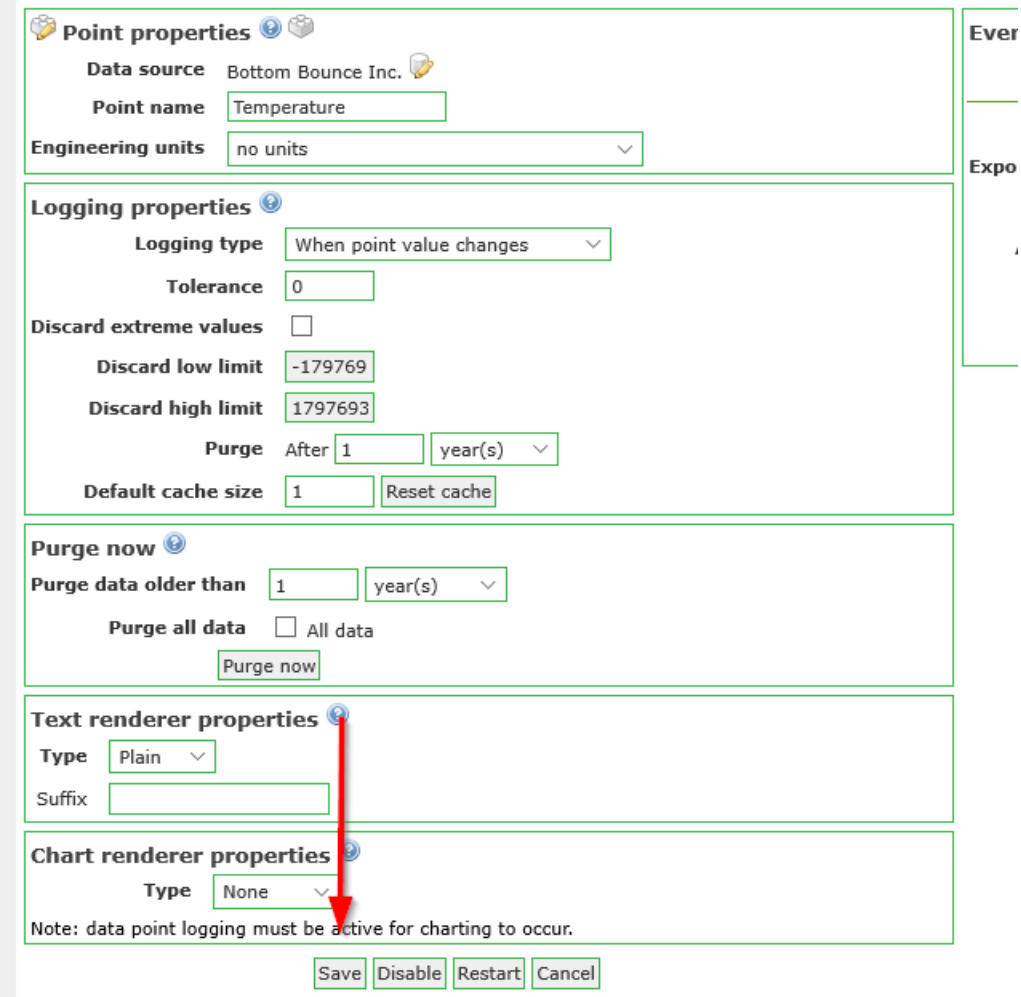
9) The **Point details** dashboard shows you a variety of information pertaining to your data source readings, such as its history, when it started, the average value (temperature), and so on. You can also view a graph representation of how the temperature (in our case) fluctuated over the course of time. This is great for administrators to ensure that the equipment does not have any abnormalities and to get more information about the data source. Feel free to explore around but do not edit anything yet.

10) Once you are done exploring around, we will refine our temperature gage a bit more, such as creating an alarm for when the temperature is too high or too low. Click on **Edit data point.**

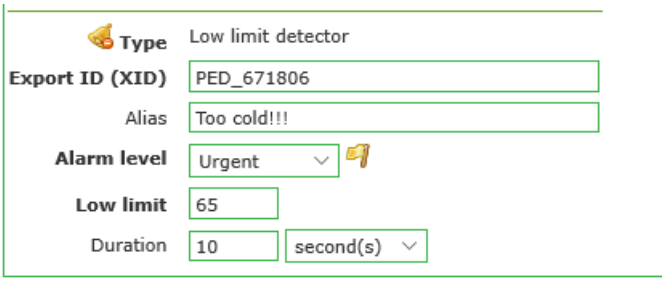


11) On the **Edit data point** dashboard, we will set upper and lower temperature alarms. Starting with the **High limit**, select **High limit** in the Type drop down menu, and click on **Add**. Ensure your inputted values mirror what is in the image below, and then click **Save**.

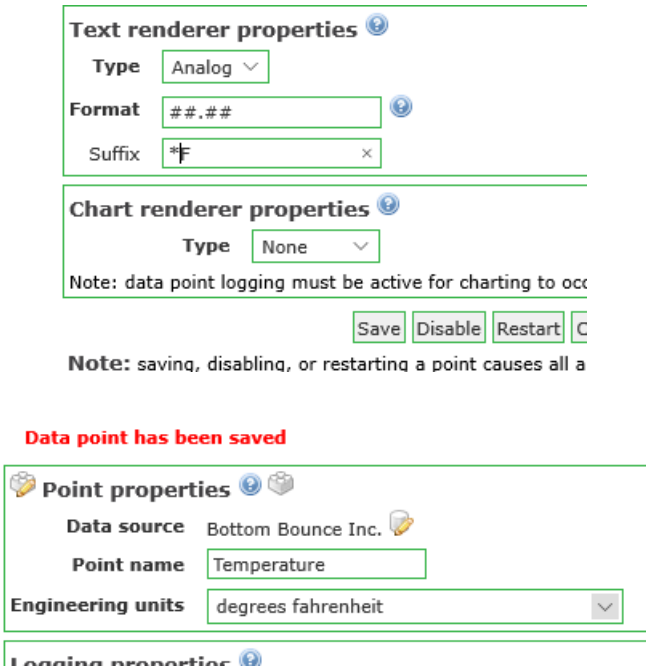




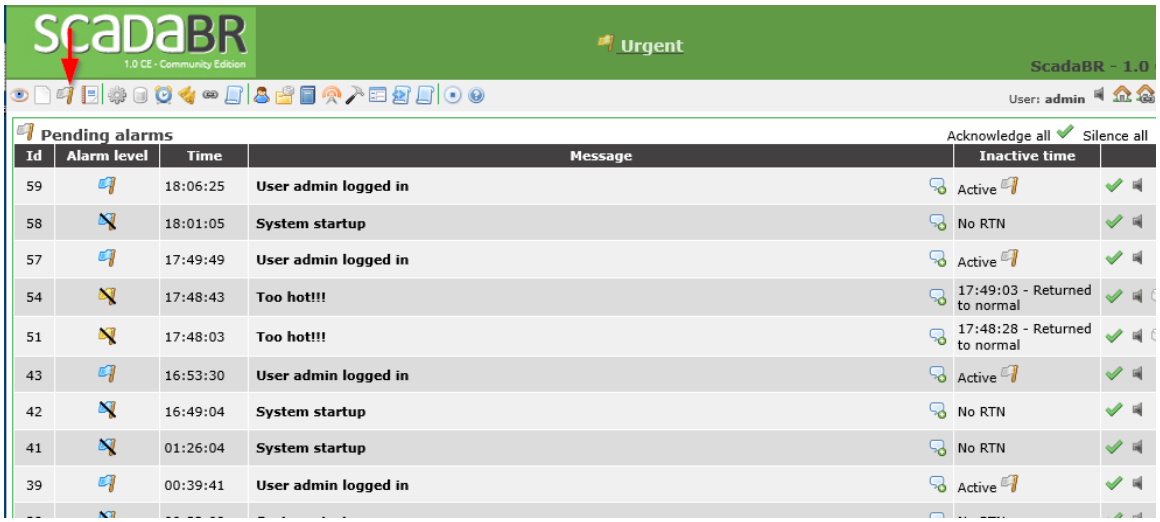
12) Do the same for the **Low limit** detector using the values below, and click **Save**



13) To clean up our temperature readings, edit the **Text renderer properties** values as seen in the image below. This will ensure that the temperature values are displayed as ##.##*F (degrees Fahrenheit). Additionally, select the Engineering units as **degrees Fahrenheit** in the drop down menu underneath Point properties. Ensure you click **Save** afterwards.



14) Once complete with the steps above, click on the **Alarms** icon (flag) to see the different alarms for the data source. Notice that the alarms include the high and low event detectors, as well as the different user logins. You can also click on the **Point details** next to the event alarms for more details.



Id	Alarm level	Time	Message	Inactive time
59		18:06:25	User admin logged in	Active
58		18:01:05	System startup	No RTN
57		17:49:49	User admin logged in	Active
54		17:48:43	Too hot!!!	17:49:03 - Returned to normal
51		17:48:03	Too hot!!!	17:48:28 - Returned to normal
43		16:53:30	User admin logged in	Active
42		16:49:04	System startup	No RTN
41		01:26:04	System startup	No RTN
39		00:39:41	User admin logged in	Active

15) Exit out of Internet Explorer, and exit the VM, ensuring you select **Save the machine state**.

MISSION OVERVIEW

Note: the following labs require a basic understanding of how networks operate, basic knowledge of Linux commands, and basic knowledge of Wireshark. For a better learning experience, I **highly** recommend you to brush up on those topics before proceeding. Additionally, all the **bolded** parts of a command represent a value that pertains to your particular machine.

"Welcome to the team," said your boss Admiral Jimbo McFly as he sat on his office chair and lit up a cigar.

"Bottom Bounce Inc. has been involved in some shady business lately, and it is your job to stop them. I won't go into much detail as this is a Top Secret mission, but your goal is to figure out ways to get into their SCADA/ICS system and wreak havoc - all while undercover. They do not take physical security very seriously over there, so you will be disguised as a janitor going by the name of Dilbert Flabbergaster. Once in the facility, you will get into their wireless network, and do what you gotta do. The wireless network is called **Bottom Bounce Inc.** and the password is **machomanrandysavage123**. Intelligence tells us that they don't really check their system logs and assume that whoever is in their network must work there. Intelligence also tells us that there should only be three machines in the network: a Windows 10 used for remote administration of the ScadaBR server, the ScadaBR server itself, and you. Any questions?"

"No sir," you reply as you start to get ready to proceed to Bottom Bounce Inc.'s SCADA/ICS facility.

LAB 1: RECONNAISSANCE

Just like navigating from point A to point B on a road trip, the first step is figuring out our location. In offensive cyber operations, this means finding out our IP address in the network and which network we are in. After knowing where we are at in time and space, we proceed to figure out what else is around us and if there is anything of interest. This is normally (if not always) done **passively** fist so as to not alert our target that we are attempting to exploit their network/systems. However, for the sake of simplicity and other reasons beyond the scope of this lab (i.e. running Wireshark in promiscuous mode), we will start off with an **active** reconnaissance of our network. Let's fire up our VMs:

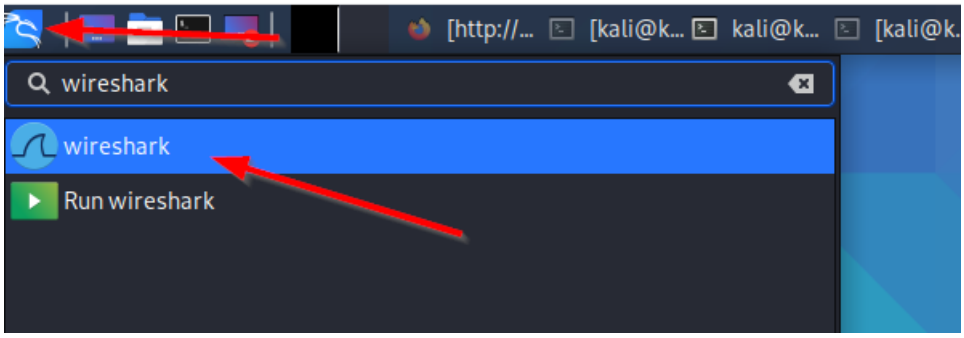
1) Open your *Oracle VM VirtualBox Manager*.
2) **Start** your Kali VM
3) **Start** your ScadaBR VM
4) **Start** your Windows 10 VM
5) On your Kali VM, open up a terminal and type the following (if prompted for the password, the password is *kali*):
    - *sudo ifconfig*
6) If successful, you should see the following on the terminal, although your IP may be different:

```
kali@kali:~$ sudo ifconfig
[sudo] password for kali:
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.21  netmask 255.255.255.0  broadcast 10.0.3.255
```

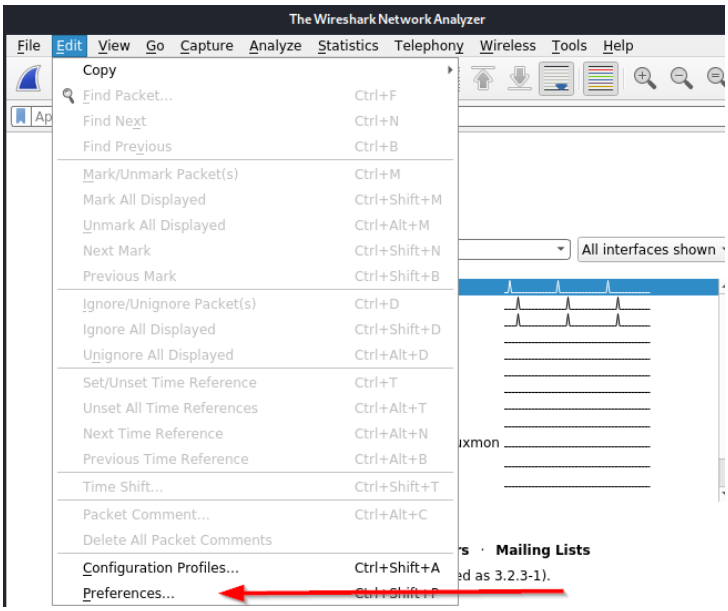
7) Based on this information, we can determine our IP address and the network address our Kali VM is in. In my case, my IP address is *10.0.3.21* and the network IP address is *10.0.3.0/24* as the netmask is *255.255.255.0*.
8) With this information, we will now conduct a basic nmap ping scan on our network to find out which hosts are up. To do so, go to your *bottombounce* folder (found on your desktop), open a terminal (right click > *Open Terminal Here*), and type the following:
    - *nmap –sn **10.0.3.0/24** > pingscan.txt*
9) If all went well, you should see something similar to the following after using the *cat pingscan.txt* command:

```
kali@kali:~/Desktop/bottombounce$ cat pingscan.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-13 23:58 EDT
Nmap scan report for 10.0.3.21
Host is up (0.00030s latency).
Nmap scan report for 10.0.3.24
Host is up (0.00068s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 3.04 seconds
```

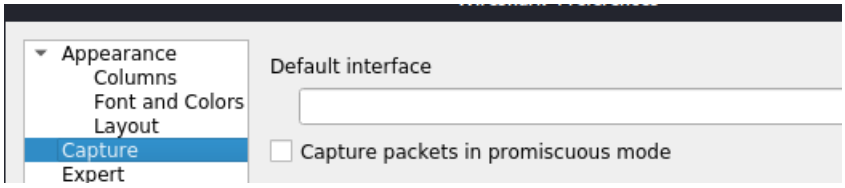
10) As you can see above, there are two addresses up: *10.0.3.21* and *10.0.3.24*. We know that *10.0.3.21* (or whatever your Kali's IP is) is our Kali VM, but what about the *10.0.3.24*? And what about the Windows 10 VM we setup – what is its IP address?
11) First, let's try to figure out what the Windows 10 IP address is. Open up Wireshark on your Kali VM by clicking on the blue Kali icon on the top left corner of your Kali VM > *09 – Sniffing & Spoofing* > *wireshark*. Another option is to type *wireshark* on the searchbox as shown below.



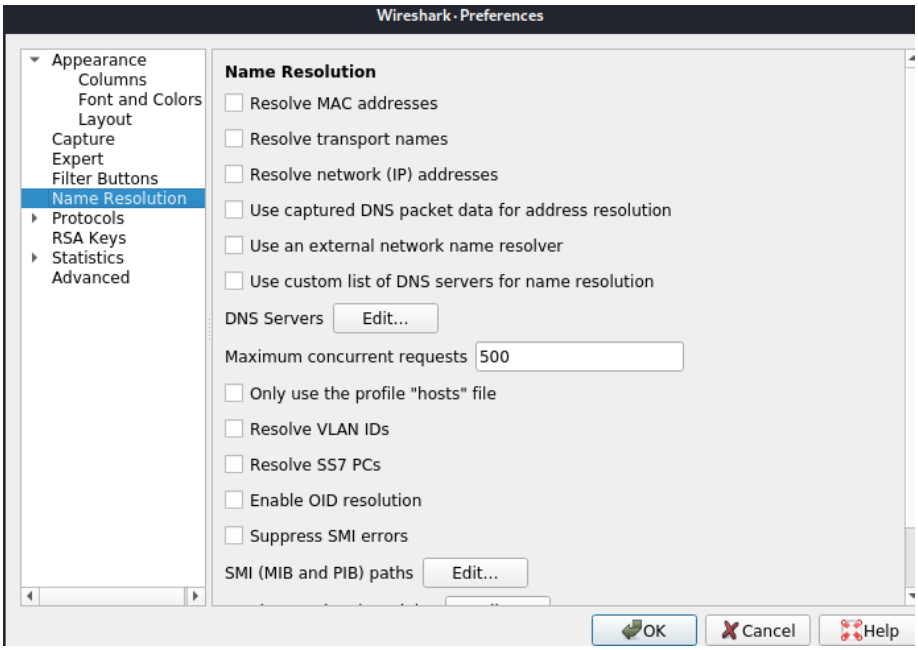
12) If prompted for a password, just type in *kali* per usual. This will land you on Wireshark's main dashboard. To keep things more "realistic" and more efficient, we will make some modifications to our capture.
    - Select *Edit* > *Preferences*



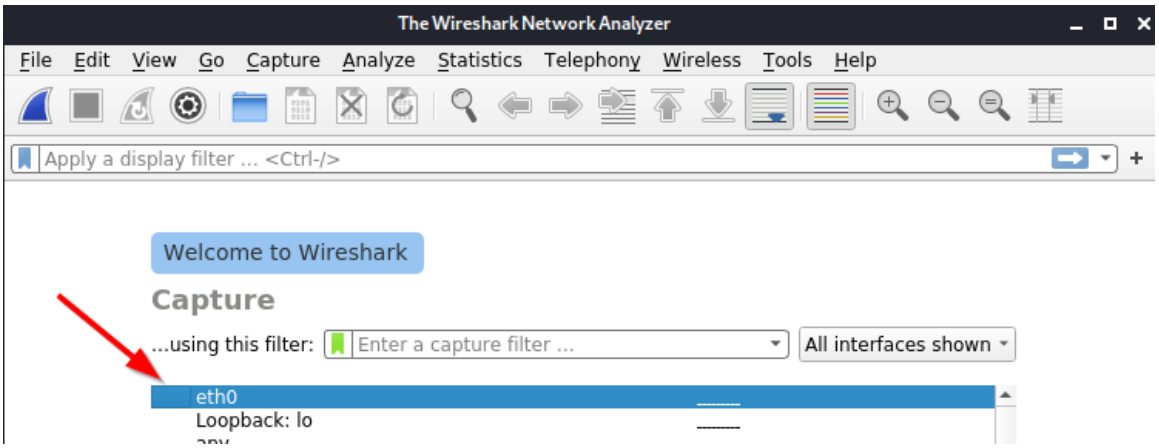
- Now click on *Capture* and **uncheck** the "Capture packets in promiscuous mode". This will ensure that Wireshark is not capturing any conversations outside of what is being sent to the Kali VM and to what is being broadcasted.



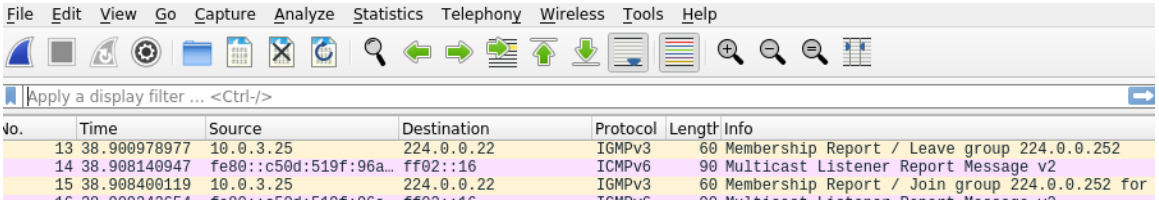
- Now click on *Name Resolution* and uncheck all the boxes. This will eliminate some of the clutter associated with packet capturing using a VM.



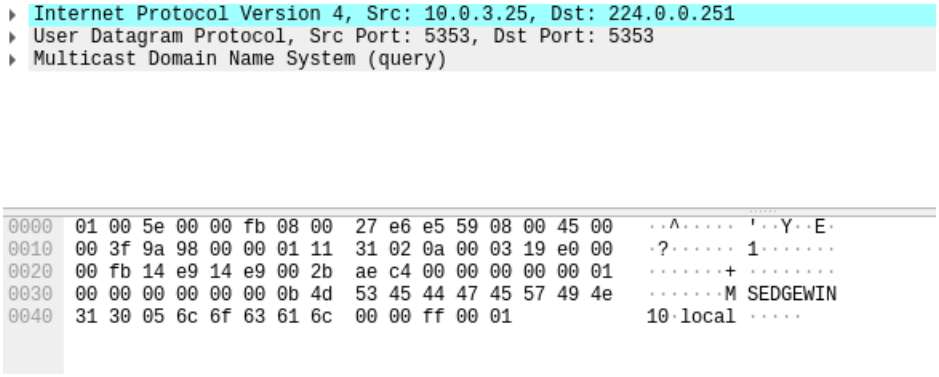
- Once all the changes are done, click *Ok* and **exit** out of Wireshark.
- Now open Wireshark up back again, and double click on *etho* (or whatever Kali's interface is being used in accordance with your earlier *ifconfig* display).



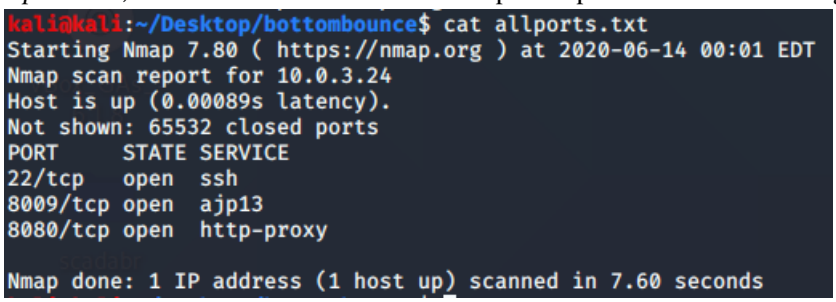
- Now wait and see what happens. If you payed close attention before double clicking *etho*, you might have noticed some spikes, indicating that there is traffic in the network (even if you didn't generate any). If you wait about a minute or so, you should see some traffic populate Wireshark and see some IP addresses from our network not listed on our nmap scan, as seen in the image below:



- Click through these packets to see if you can get more useful (unencrypted) information. After clicking on a MDNS packet sent from the *10.0.3.25* machine, we can see that the *10.0.3.25* IP address belongs to the Windows 10 VM as seen in the image below. Without going into too much detail, one of the reasons why our Windows 10 VM did not show up in our nmap scan is because it is normally configured to block ping (ICMPv4 echo requests) probes. If interested, you can find more information here: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-firewall/create-an-inbound-icmp-rule



13) Putting it all together, we determined the following:
    - Bottom Bounce SCADA/ICS Network: 10.0.3.0/24
    - Kali IP: 10.0.3.21
    - Windows 10 IP: 10.0.3.25
    - ScadaBR: 10.0.3.24???
14) The answer is yes – the 10.0.3.24 belongs to the ScadaBR server. However, for the sake of this lab, let's ensure that that is the case. Go to your *bottombounce* folder, and open a terminal. On the terminal, type the following (T4 = fast scan, where 5 is fastest and 0 is slowest; -p- = all ports):
    - *nmap –T4 –p- 10.0.3.24 > allports.txt*
15) If we cat *allports.txt*, we can see that there are three ports open as seen in the image below



16) Although that does not tell us exactly that the IP belongs to the ScadaBR server, we can still see that ports 22, 8009, and 8080 are open. Port 22, specifically, should certainly peak our interest as we could get root access to the machine with the proper credentials (or brute forcing, as we will see in a future lab). Port 22 also indicates some need for remote administration to be conducted. Port 8080, on the other hand, tells us that there is some

sort of http website being run. Let us dig deeper into these ports – type the following in the terminal (-A = "all" information possible, if you will:

- *nmap –T4 -A –p 22,8009,8080 10.0.3.24 > juicyports.txt*

17) After *cat*ting *juicyports.txt*, we should see more information pertaining to ports 22, 8009, and 8080 as seen in the image below. Notice that port 8080 allows PUT methods to be executed. For the sake of simplicity, this implies that users are able to modify data in this machine's hosted http website, which gives another lead that this may be in fact our ScadaBR server. For more information on PUT and other http methods, please visit https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#:~:text=The%20PUT%20method%20requests%20that,residing%20on%20the%20origin%20server.

```
kali@kali:~/Desktop/bottombounce$ cat juicyports.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-14 00:11 EDT
Nmap scan report for 10.0.3.24
Host is up (0.00044s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4p1 Debian 10+deb9u3 (protocol 2.0)
| ssh-hostkey:
|   2048 1e:01:9a:c1:fc:29:56:71:b2:3e:5b:50:c6:cc:d2:e2 (RSA)
|   256 3e:68:b7:86:4f:de:f2:61:3a:d8:90:99:e3:a5:6f:e9 (ECDSA)
|_  256 9e:86:97:c4:d9:1c:ff:a4:0c:a4:ad:02:e5:f8:f2:bf (ED25519)
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
| ajp-methods:
|   Supported methods: GET HEAD POST PUT DELETE OPTIONS
|   Potentially risky methods: PUT DELETE
|_  See https://nmap.org/nsedoc/scripts/ajp-methods.html
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
| http-methods:
|_  Potentially risky methods: PUT DELETE
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.40 seconds
```

18) Putting all of this information together, we can infer that the *10.0.3.24* machine is running some sort of http webserver using Apache Tomcat, and that the machine requires some sort of remote administration to be conducted via port 22 (ssh). So, using our mission overview intelligence, we may come to the conclusion that this is indeed the ScadaBR server's IP address.