Malware Development for Red Teaming

November 05, 2021

Welcome to the Malware Development workshop for AfricaHackon 2021. In this interactive workshop, we will take a look at the C# language and how to write malware focused on droppers/loaders that will run shellcode on Windows 10 targets that give a meterpreter session back.

The flow of this workshop is building out the code like how virus research looks at ***Gain of Function***. We start off with a simple shellcode runner and build out from there.

In order to be able to follow the workshop properly, it is highly recommended to go through **Lab 0** before the start of the workshop, as this is a setup lab to get you up and ready.

All commands to be executed in Kali will be in **red** and **bold** while all commands to be executed in the Development Environment will be in **purple** and **bold**.

Lab0: Setup

Download Virtualization Software like VMware Workstation Player (https://customerconnect.vmware.com/en/downloads/info/slug/desktop_end_user_computing/vmware_workstation_player/16_0) or VirtualBox (https://www.virtualbox.org/wiki/Downloads)

Note: *Although both work, the labs in this guide were prepared and tested using VirtualBox.*

Download the following Operating Systems:

1. Windows 10 image which will be the Development Environment (https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise)

   For this machine, the recommendation is to allocate at least 4 GBs of RAM and 2 CPUs and 50GB disk space.

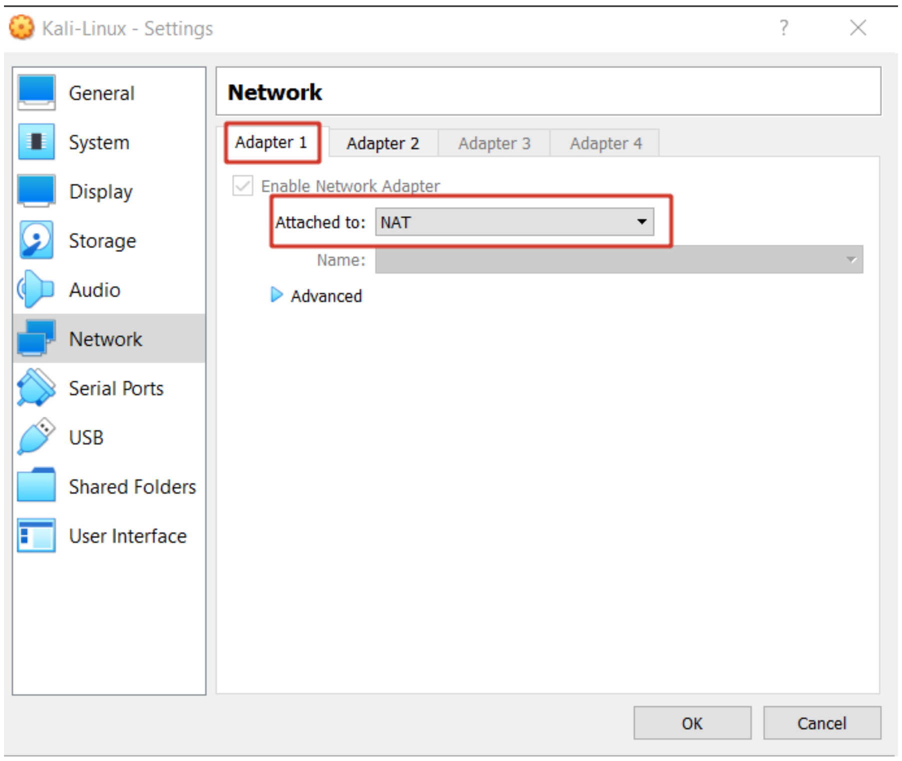
2. Kali Linux image (https://kali.download/virtual-images/kali-2021.3/kali-linux-2021.3-virtualbox-amd64.ova)

   For this machine, the recommendation is to allocate at least 4 GBs of RAM and 2 CPUs and 25GB disk space.

Note: *I assume that the reader is able to do base installations of the above OSs in Virtualbox.*

Virtualbox Setup

To set up the lab we need to initially have internet connection to all machines so that we can download any necessary software. To do this I configured each VM with its first Network Adapter Settings set as **NAT**.


Kali-Linux - Settings

Network

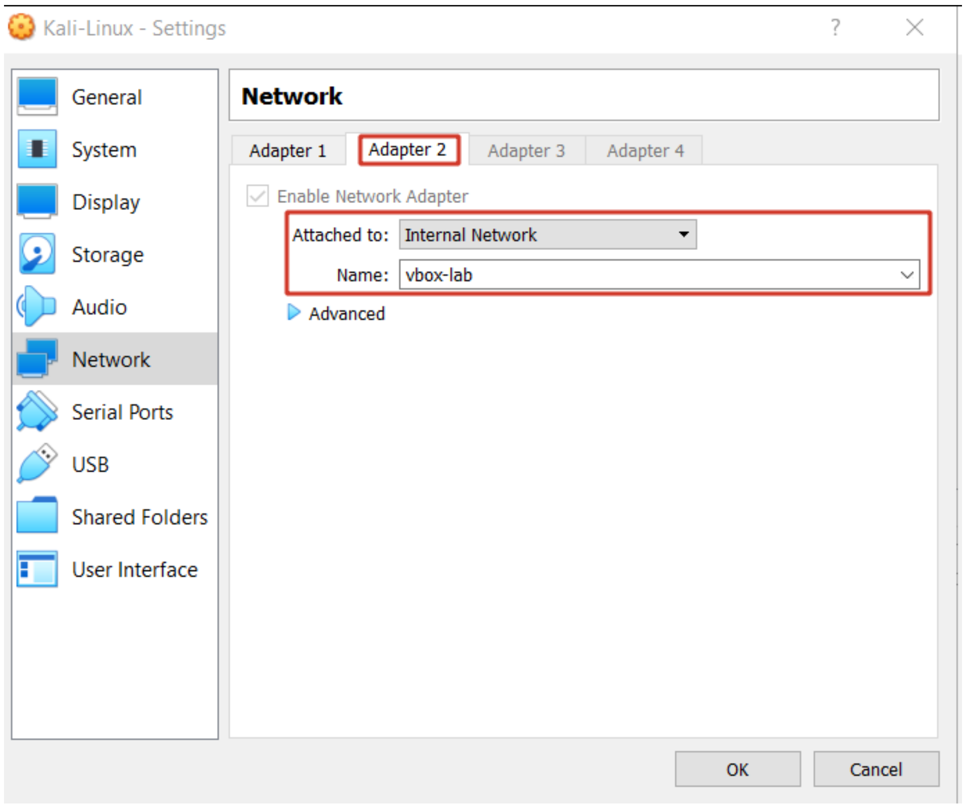
Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

Enable Network Adapter

Attached to: NAT

Name:

Also considering that there VMs will need to communicate with each other I configured the second adapter as **Internal Network** and give it a name that I can easily recall like **vbox-lab**.


Kali-Linux - Settings

Network

Adapter 2

Enable Network Adapter

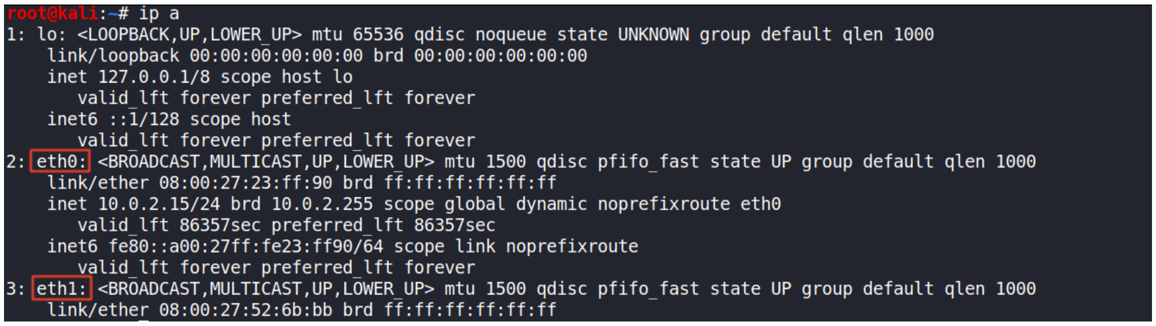
Attached to: Internal Network

Name: vbox-lab

Kali Linux Setup

The main thing that will be needed to be configured on the Kali VM is the network. Log in to the Kali machine and running this in terminal:

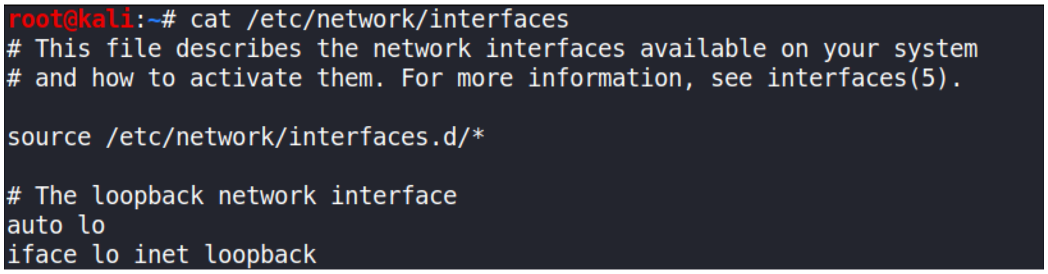
```
ip a
```



```
root@kali:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:23:ff:90 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 86357sec preferred_lft 86357sec
    inet6 fe80::a00:27ff:fe23:ff90/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:52:6b:bb brd ff:ff:ff:ff:ff:ff
```

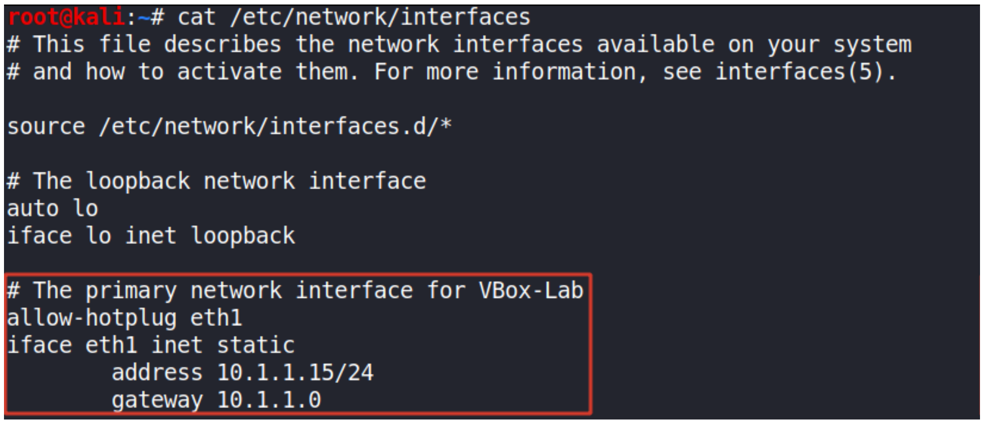
The two network adapters can be seen. **eth0** is the **NAT** network giving internet access while **eth1** is the **vbox-lab** network. We have to assign a static IP to eth1. To do this we need to modify the **interfaces** file located at **/etc/network/**. Running the following command should give a similar output as shown in the screenshot below.

cat /etc/network/interfaces



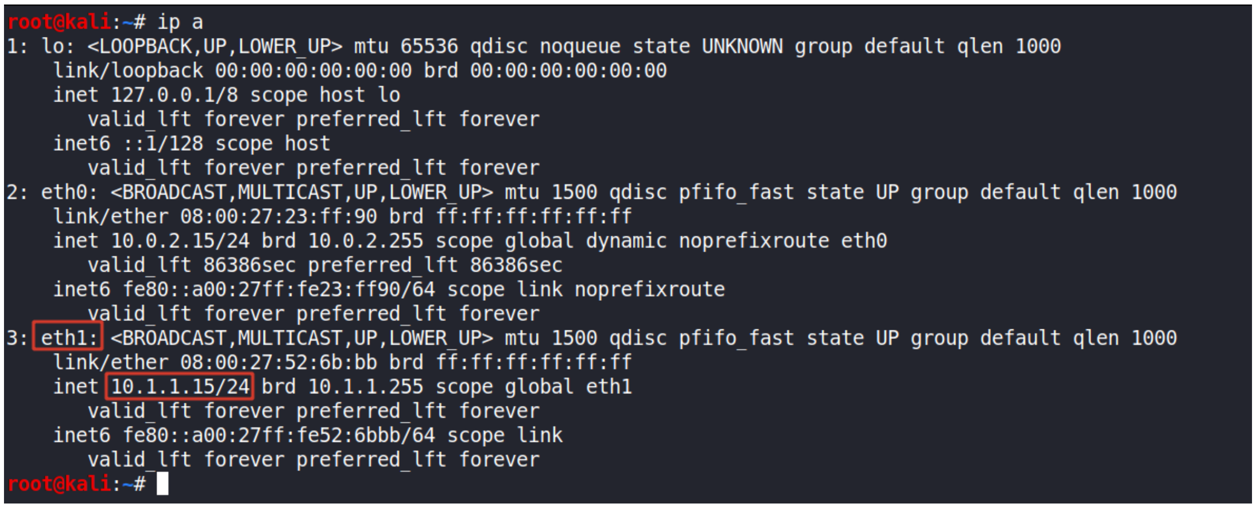
```
root@kali:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback
```

Using vim or your favorite editor, modify the contents of interfaces to assign the Kali machine the ip of **10.1.1.15** and the full network mask of /24 and specify the gateway of **10.1.1.0**. The file should then look like this:



```
root@kali:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface for VBox-Lab
allow-hotplug eth1
iface eth1 inet static
        address 10.1.1.15/24
        gateway 10.1.1.0
```

Once you restart the network service the interface will get assigned the ip and look like this:

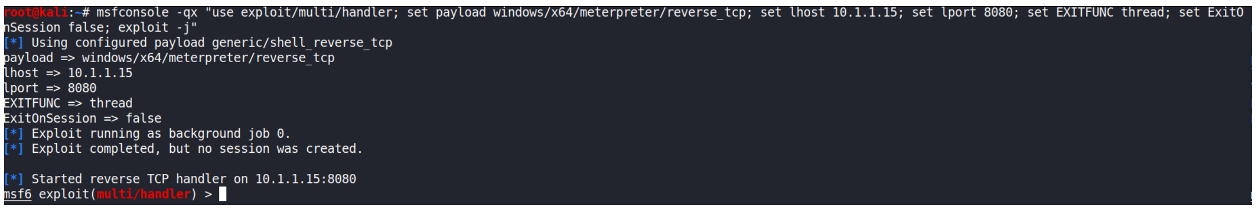


```
root@kali:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:23:ff:90 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 86386sec preferred_lft 86386sec
    inet6 fe80::a00:27ff:fe23:ff90/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:52:6b:bb brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.15/24 brd 10.1.1.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe52:6bbb/64 scope link
       valid_lft forever preferred_lft forever
root@kali:~# 
```

Next ensure that **Postgresql** service is started and **Metasploit** is functional.

sudo service postgresql start

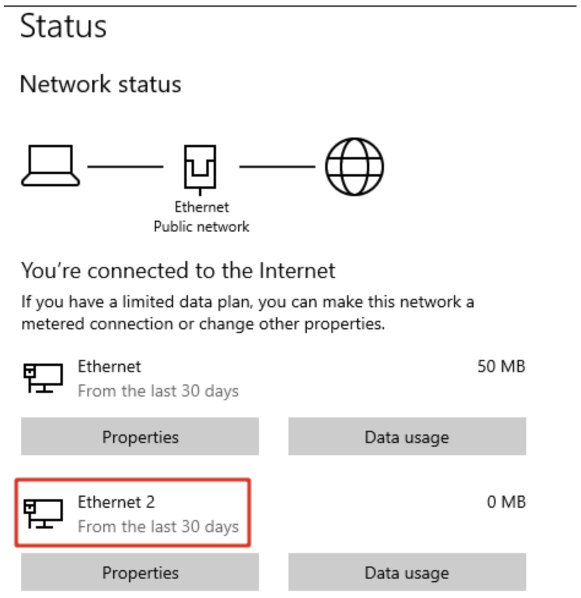
msfconsole -qx "use exploit/multi/handler; set payload windows/x64/meterpreter/reverse_tcp; set lhost 10.1.1.15; set lport 8080; set EXITFUNC thread; set ExitOnSession false; exploit -j"



```
root@kali:~# msfconsole -qx "use exploit/multi/handler; set payload windows/x64/meterpreter/reverse_tcp; set lhost 10.1.1.15; set lport 8080; set EXITFUNC thread; set ExitO
nSession false; exploit -j"
[*] Using configured payload generic/shell_reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
lhost => 10.1.1.15
lport => 8080
EXITFUNC => thread
ExitOnSession => false
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.1.1.15:8080
msf6 exploit(multi/handler) >
```

Windows 10 Setup

On the Windows 10 machine there are a number of things we will need to do. First off will be the network configuration for the **vbox-lab** adapter which will be **Ethernet 2**.



Status

Network status

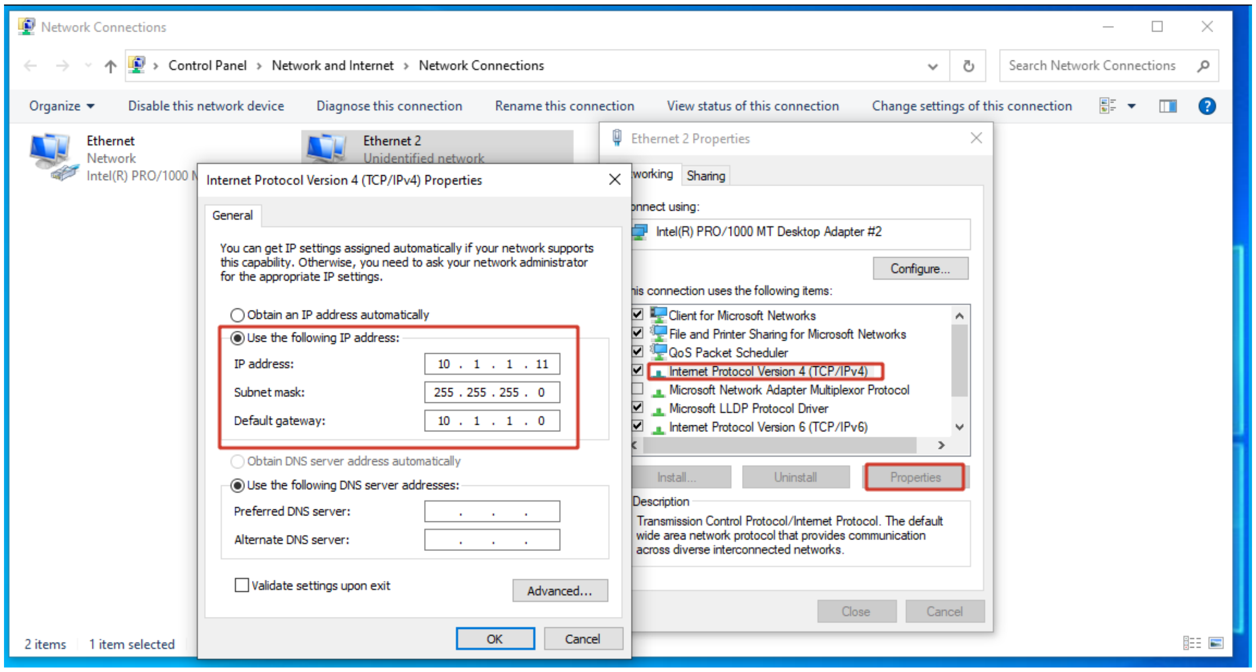
You're connected to the Internet

If you have a limited data plan, you can make this network a metered connection or change other properties.

Ethernet 50 MB
From the last 30 days

Ethernet 2 0 MB
From the last 30 days

Open up Ethernet 2 properties and configure the IP4 settings to look like this.



Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

( ) Obtain an IP address automatically
(●) Use the following IP address:

IP address:	10 . 1 . 1 . 11
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	10 . 1 . 1 . 0

( ) Obtain DNS server address automatically
(●) Use the following DNS server addresses:

Preferred DNS server:	. . .
Alternate DNS server:	. . .

☐ Validate settings upon exit

onnect using:
Intel(R) PRO/1000 MT Desktop Adapter #2

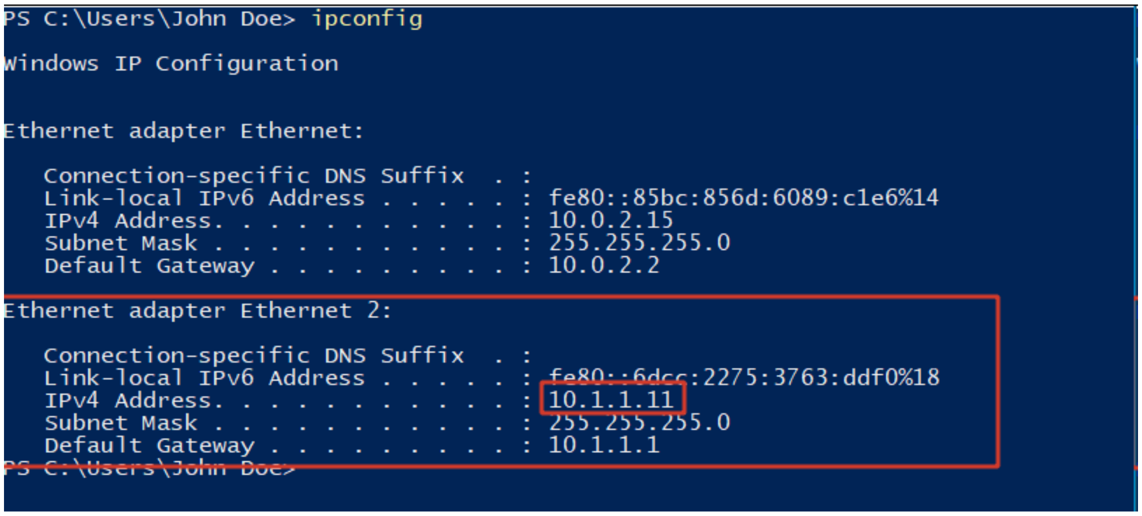
his connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☐ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Once you click ok the network adapter should inherit the configurations supplied. Verify this by opening powershell and running:

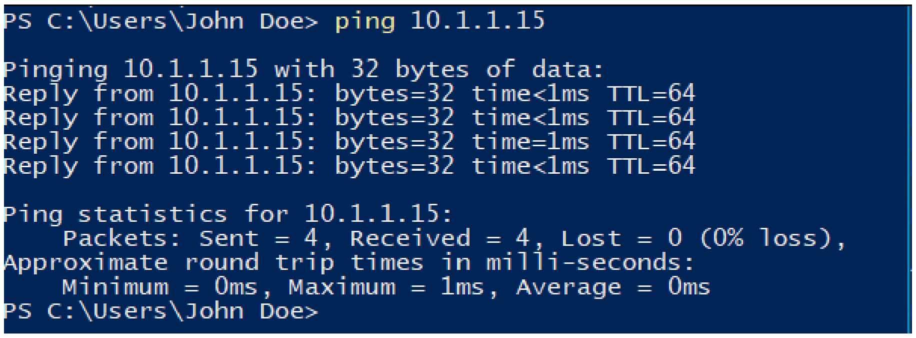
ipconfig



```
PS C:\Users\John Doe> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::85bc:856d:6089:c1e6%14
   IPv4 Address. . . . . . . . . . . : 10.0.2.15
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.2.2

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::6dcc:2275:3763:ddf0%18
   IPv4 Address. . . . . . . . . . . : 10.1.1.11
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.1.1
PS C:\Users\John Doe>
```

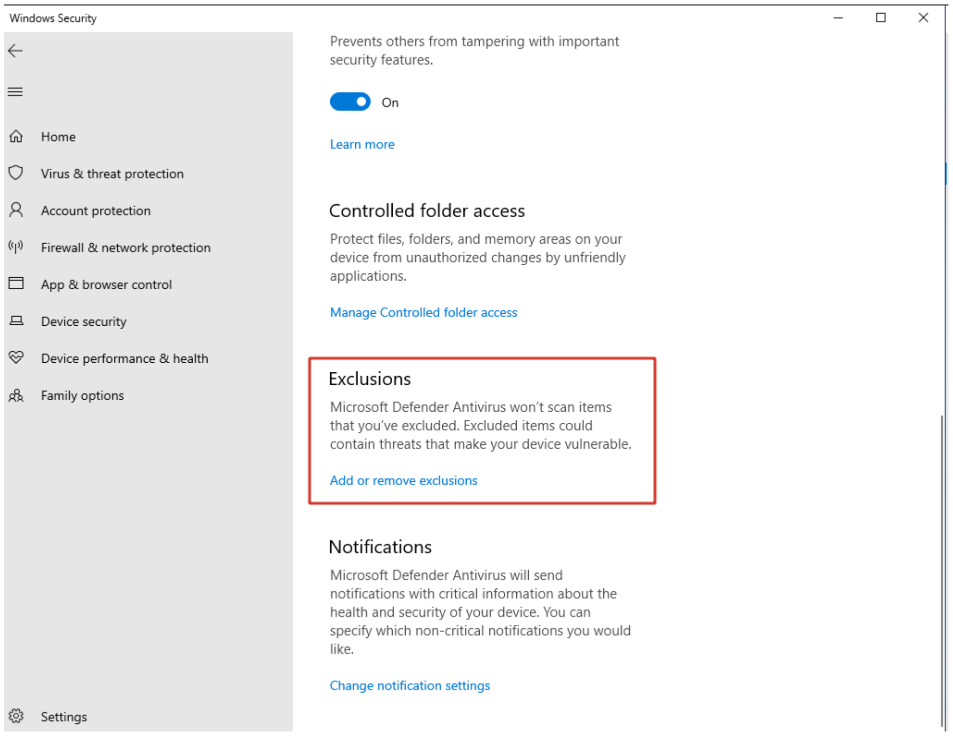
Test to see that the Development Environment can ping the Kali Vm.

ping 10.1.1.15



```
PS C:\Users\John Doe> ping 10.1.1.15

Pinging 10.1.1.15 with 32 bytes of data:
Reply from 10.1.1.15: bytes=32 time<1ms TTL=64
Reply from 10.1.1.15: bytes=32 time<1ms TTL=64
Reply from 10.1.1.15: bytes=32 time=1ms TTL=64
Reply from 10.1.1.15: bytes=32 time<1ms TTL=64

Ping statistics for 10.1.1.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\Users\John Doe>
```

Now that we have the network up and running we can proceed with configuration of an exclusion folder where we will store all our code in so that Defender will not scan them. To this create a folder on the desktop and name it **Weaponized**. Open Windows Security settings and navigate to **Exclusions**. Click on the **Add or remove exclusions** option



Prevents others from tampering with important security features.

On

Learn more

Controlled folder access

Protect files, folders, and memory areas on your device from unauthorized changes by unfriendly applications.

Manage Controlled folder access

Exclusions

Microsoft Defender Antivirus won't scan items that you've excluded. Excluded items could contain threats that make your device vulnerable.

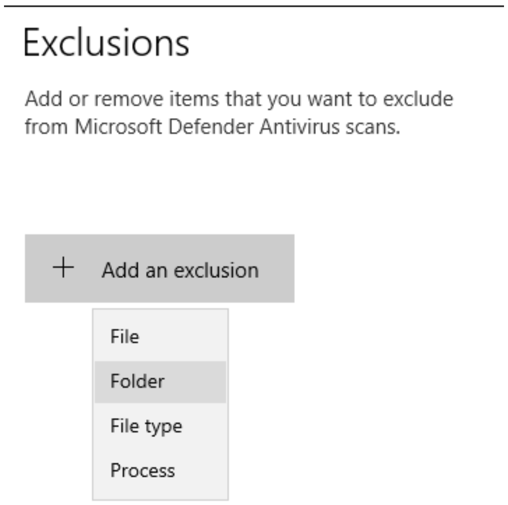
Add or remove exclusions

Notifications

Microsoft Defender Antivirus will send notifications with critical information about the health and security of your device. You can specify which non-critical notifications you would like.

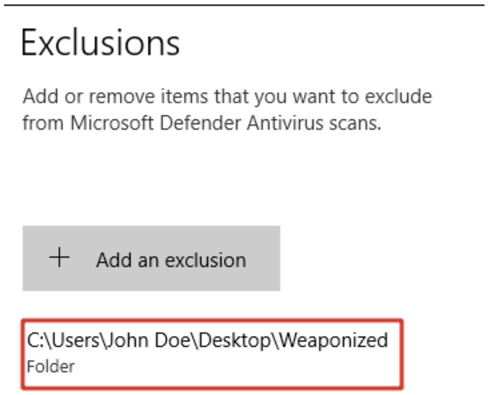
Change notification settings

Add the **Weaponized** folder that we created on the Desktop into the Exclusions list.


Exclusions

Add or remove items that you want to exclude from Microsoft Defender Antivirus scans.

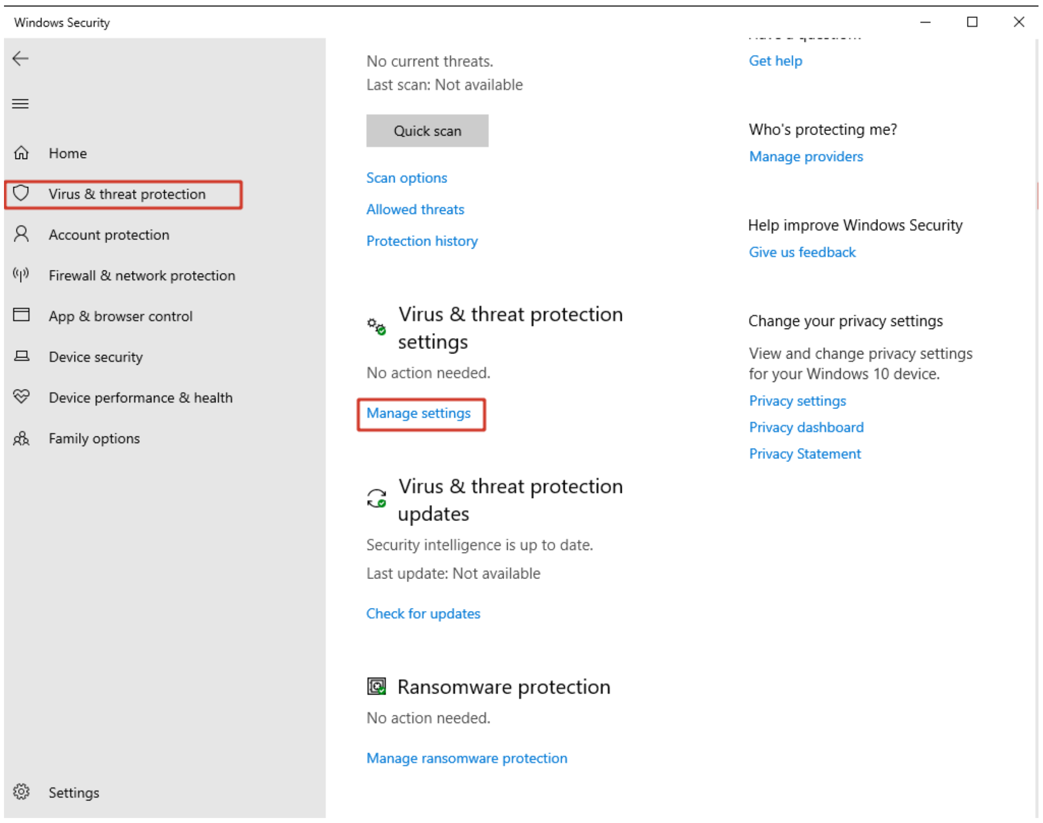
Once you have selected the appropriate folder it should look like this:


Exclusions

Add or remove items that you want to exclude from Microsoft Defender Antivirus scans.

C:\Users\John Doe\Desktop\Weaponized
Folder

Another key thing to do is disable the **Sample Submission** option in Defender so that the compiled binaries we create are not shipped off for further analysis. Again open Windows Security settings and navigate to **Virus and threat protection** and then click **Manage settings**.


Windows Security

No current threats.
Last scan: Not available

Quick scan

Scan options
Allowed threats
Protection history

Virus & threat protection settings
No action needed.
Manage settings

Virus & threat protection updates
Security intelligence is up to date.
Last update: Not available
Check for updates

Ransomware protection
No action needed.
Manage ransomware protection

Get help

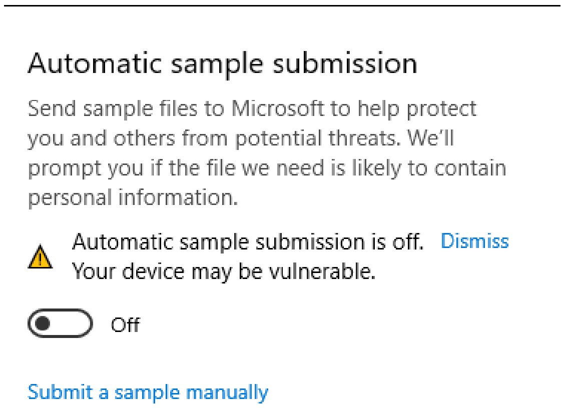
Who's protecting me?
Manage providers

Help improve Windows Security
Give us feedback

Change your privacy settings
View and change privacy settings for your Windows 10 device.
Privacy settings
Privacy dashboard
Privacy Statement

Home
Virus & threat protection
Account protection
Firewall & network protection
App & browser control
Device security
Device performance & health
Family options
Settings

Turn off the **Automatic sample submission** option and click the **Dismiss** option if you want to block alerts telling you to enable it.


Automatic sample submission

Send sample files to Microsoft to help protect you and others from potential threats. We'll prompt you if the file we need is likely to contain personal information.

Automatic sample submission is off. Dismiss
Your device may be vulnerable.

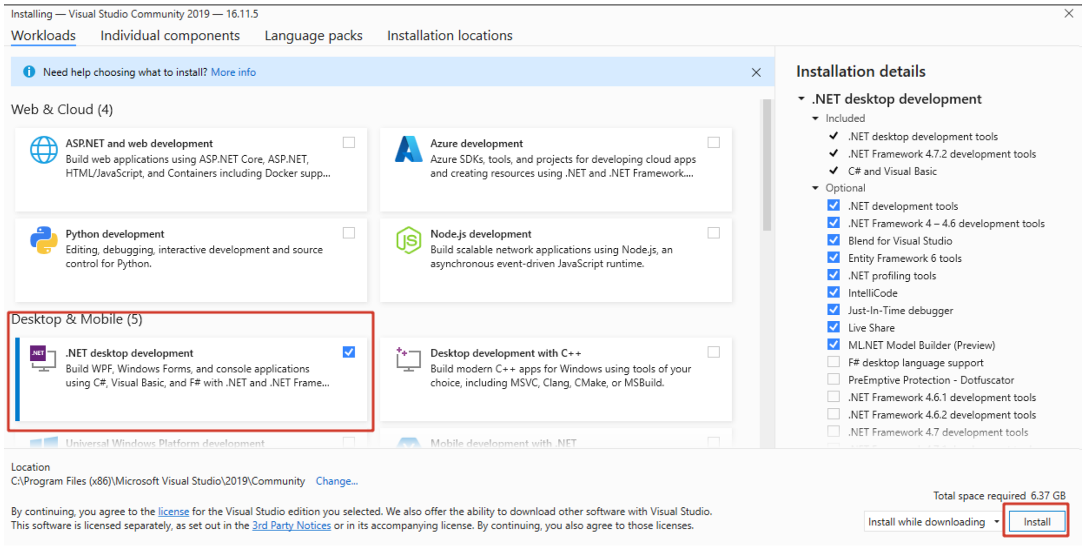
Off

Submit a sample manually

With all that complete we will need to install the following tools on the VM:

- Visual Studio 2019 Community Edition (https://visualstudio.microsoft.com/vs/community/)
- Your favorite text editor: Notepad++ (https://notepad-plus-plus.org/download), Sublime Text (https://www.sublimetext.com/3), etc.
- Process Hacker (https://processhacker.sourceforge.io/downloads.php)
- dnSpy (https://github.com/dnSpy/dnSpy/releases)
- ConfuserEx (https://github.com/yck1509/ConfuserEx/releases)
- SylantStrike (https://github.com/CCob/SylantStrike) * This will need to be cloned and compiled as there is no release available.

Note: When installing Visual Studio ensure that you install the .NET desktop development environment.



Installing — Visual Studio Community 2019 — 16.11.5

Workloads | Individual components | Language packs | Installation locations

Need help choosing what to install? More info

Web & Cloud (4)

- ASP.NET and web development — Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
- Azure development — Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....
- Python development — Editing, debugging, interactive development and source control for Python.
- Node.js development — Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

- [✓] .NET desktop development — Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...
- Desktop development with C++ — Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
- Universal Windows Platform development
- Mobile development with .NET

Installation details

.NET desktop development

Included
- ✓ .NET desktop development tools
- ✓ .NET Framework 4.7.2 development tools
- ✓ C# and Visual Basic

Optional
- ✓ .NET development tools
- ✓ .NET Framework 4 – 4.6 development tools
- ✓ Blend for Visual Studio
- ✓ Entity Framework 6 tools
- ✓ .NET profiling tools
- ✓ IntelliCode
- ✓ Just-In-Time debugger
- ✓ Live Share
- ✓ ML.NET Model Builder (Preview)
- ☐ F# desktop language support
- ☐ PreEmptive Protection - Dotfuscator
- ☐ .NET Framework 4.6.1 development tools
- ☐ .NET Framework 4.6.2 development tools
- ☐ .NET Framework 4.7 development tools

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 6.37 GB

NOTE: If you do not want to set up the Development Environment from scratch you can download this .ova file and import it into Virtualbox:

https://drive.google.com/file/d/1X7tSDEqkCbd0zZIk7bxhnveTsztA8IFs/view?usp=sharing

Windows account credentials: User: **John Doe** | Password: **secret1+**