

Malware Development for Red Teaming

November 05, 2021

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.

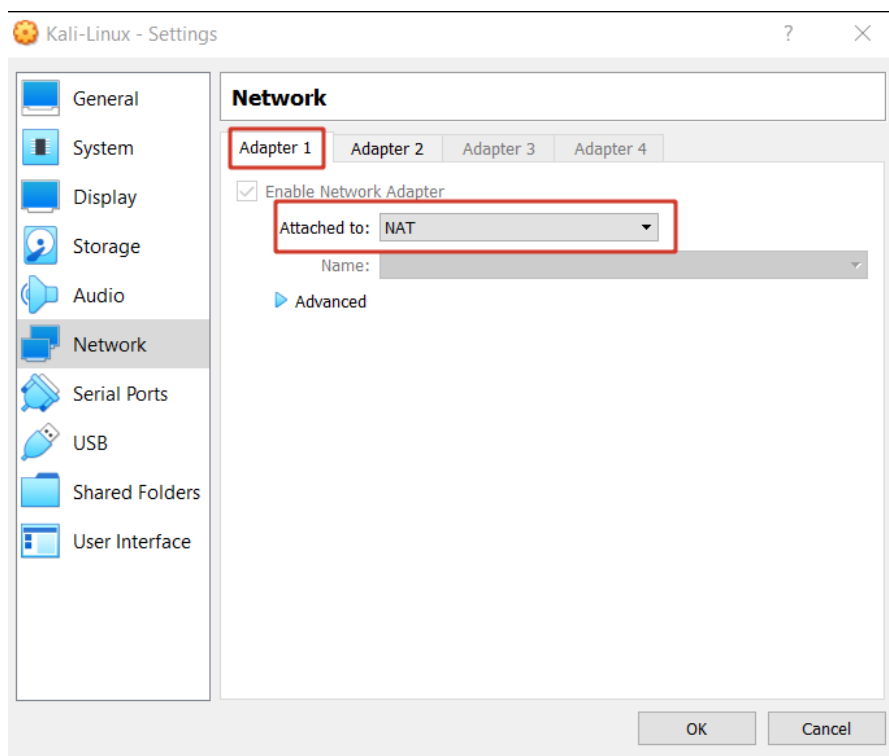
2. Kali Linux image
(<https://kali.download/virtual-images/kali-2021.3/kali-linux-2021.3-virtualbox-amd64.ova>)

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

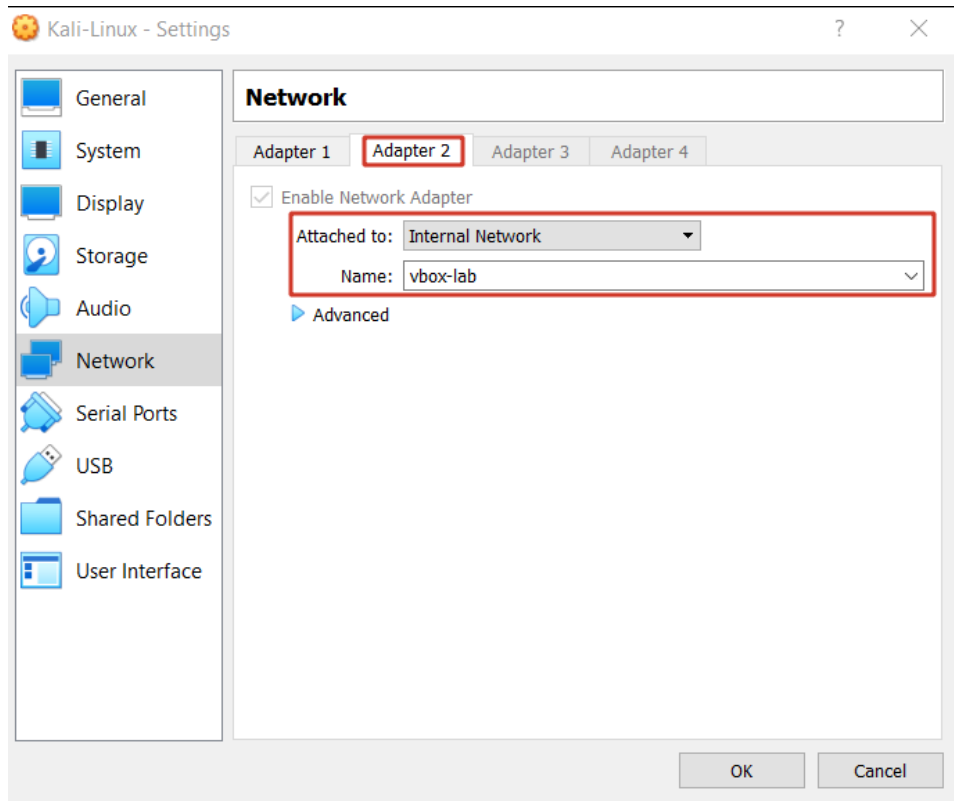
Note: I assume that the reader is able to do base installations of the above OSs in Virtualbox.

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**.



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 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 ExitOnSession 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) >

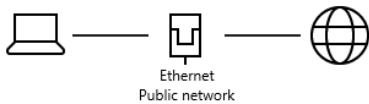
```

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

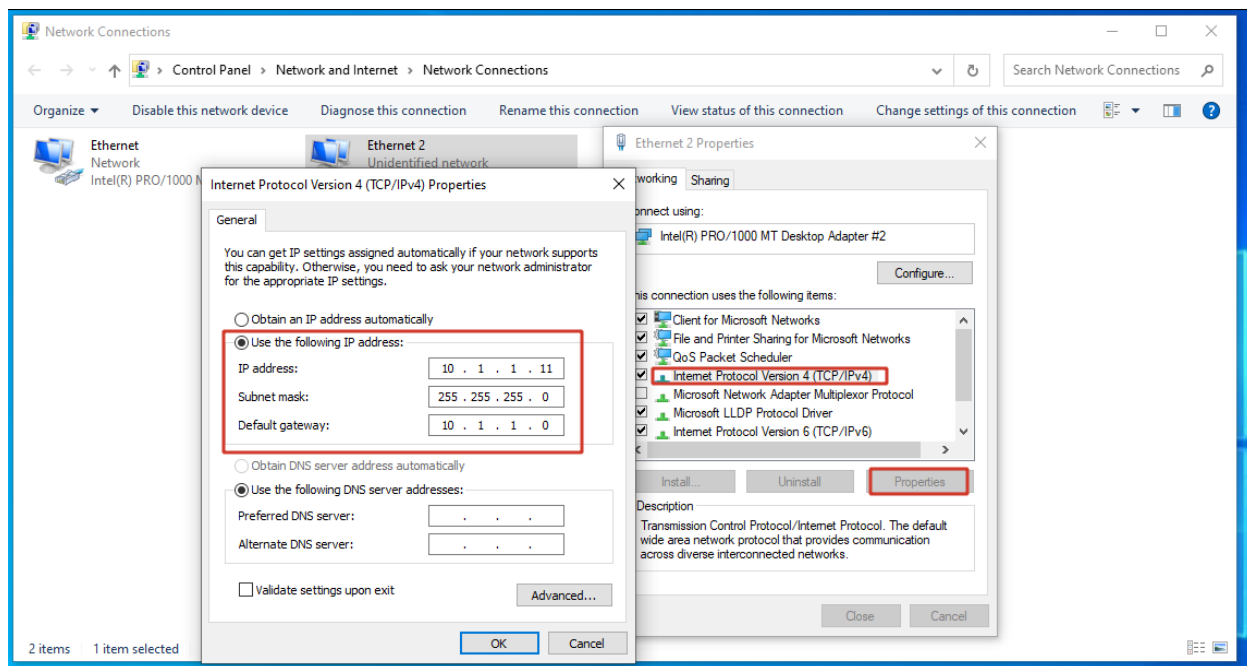


You're connected to the Internet

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

	Ethernet From the last 30 days	50 MB
Properties		Data usage
	Ethernet 2 From the last 30 days	0 MB
Properties		Data usage

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



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::6d6c: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.

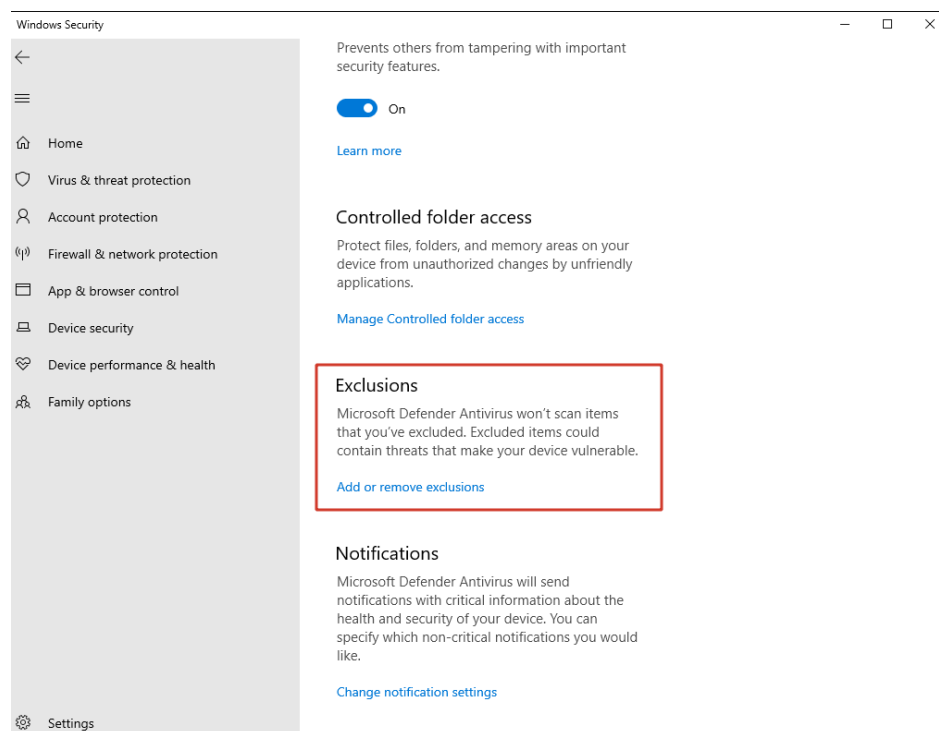
ping 10.1.1.15

```
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>
```

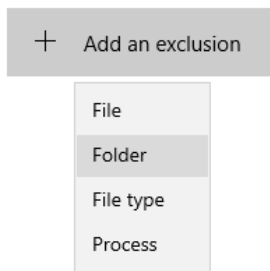
Now that we have the network up and running we can proceed with configuration of an exclusion folder where we will store all our code in so that Defender will not scan them. To this create a folder on the desktop and name it **Weaponized**. Open Windows Security settings and navigate to **Exclusions**.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



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

Exclusions

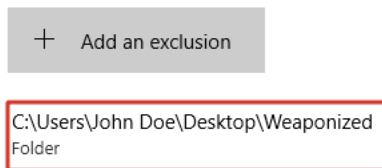
Add or remove items that you want to exclude from Microsoft Defender Antivirus scans.



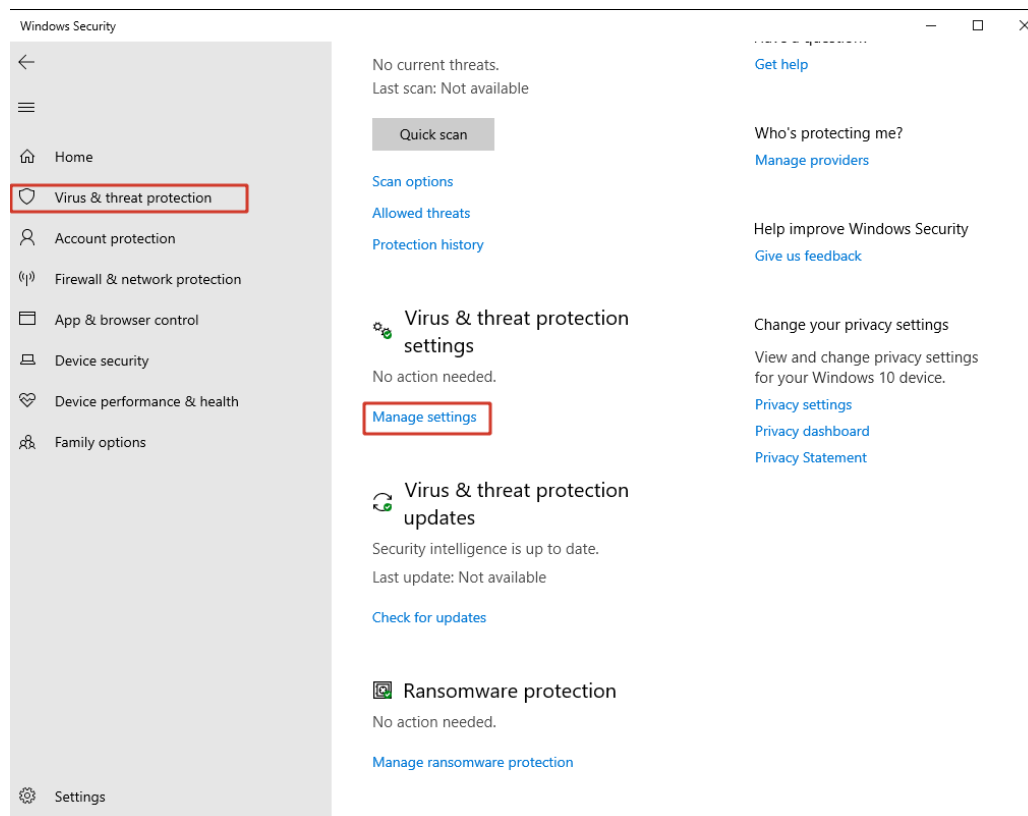
Once you have selected the appropriate folder it should look like this:

Exclusions

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




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**.



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.

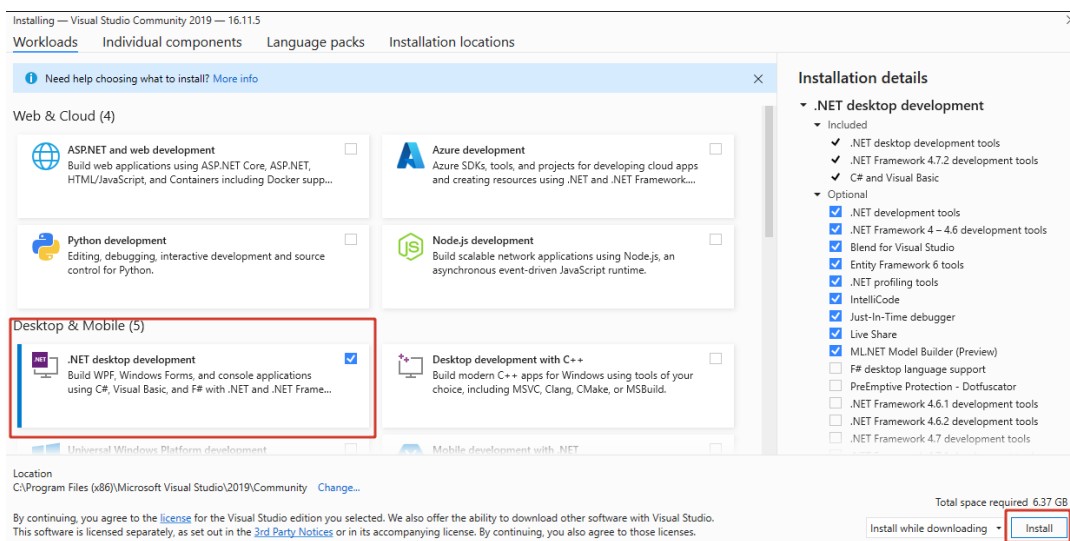
☐ Off

[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.



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+**