Analysis Sandbox Configuration

V 5.0 (27 October 2020)

LA-UR-18-29356

This doc describes how to set up an environment that will be used for the Malware Analysis Track. It can also serve as a base for your own analysis set up. Questions, comments, corrections, etc. should be sent to laurenp@lanl.gov

Required Host Laptop

- The host must be able to run two virtual machines concurrently. We recommend at least 8 gigs of RAM.
- You must have VMWare Workstation Pro (**Essential for snapshotting)**
  - VMWare Workstation Player is **not** sufficient for this workshop
  - There is a free 30 day trial of VMWare Workstation Pro available.

Required Operating Systems

- Windows 7 or 10 64 bit iso and activation key
  - Your organization's IT department may be able to help you acquire this
  - IdaFree, one of our primary tools, will only work on a 64 bit machine.
- REMnux, found at https://remnux.org/

There are many options to emulate a network. If you already have a sandnet set up that includes Windows 7 or 10 x64, you are welcome to use it (self-supported)

Building REMnux VM

1. Download the remnux ova at https://remnux.org/
    a. We recommend the google drive mirror. The ufile.io website seems to throttle download speeds.
2. In VMWare Workstation go to File → Open → remnux-6.0-ova-public.ova
3. Follow the prompts to import the .Ova
4. Power on and update the machine (`sudo apt-get update, sudo apt-get upgrade`)
5. Install Bless hex editor (`sudo apt-get install bless`)
6. Snapshot and name “clean” or something that makes sense to you

Setting Up the Windows VM

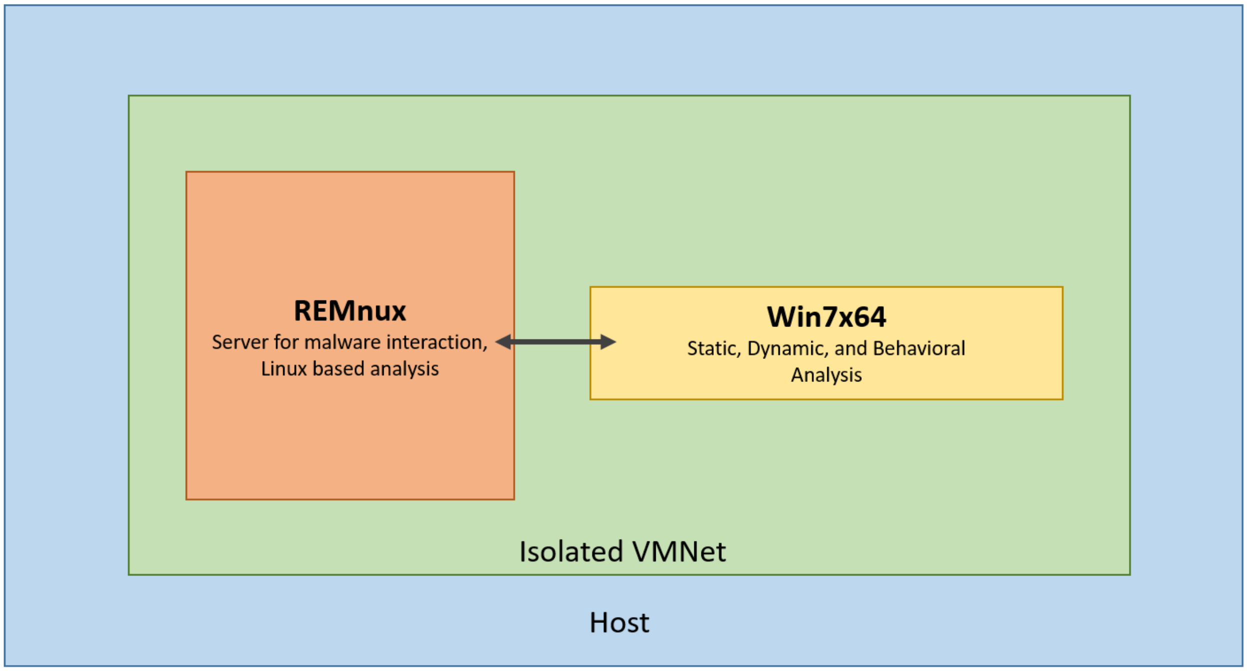
1. Install and update the OS. Take a snapshot and name it something like “Base OS Install” or something meaningful to you.
2. Install VMWare Tools - VM → Install VMWare Tools. Reboot.
    a. Note, VMWare tools now requires Service Pack 1 – you cannot complete this step if you have Windows 7 and have not updated the operating system to SP1
3. Change settings to show hidden folders/files and file extensions – My Computer → Organize → Folder and Search Options → View → check “Show hidden files…” , uncheck “Hide extensions…”
4. Make a folder C:\Reversing. This is where we will put all of our tools (those with and without installers). Make a desktop shortcut to it to save time.
5. Set a different desktop background to remind you that this is a malware VM.
6. Install the following list to the C:\Reversing directory.

    - Ida Free or Ida Demo, whichever license matches your situation best
        - Ida Free: https://www.hex-rays.com/products/ida/support/download_freeware.shtml
    - Explorer Suite http://www.ntcore.com/exsuite.php
    - 7zip http://www.7-zip.org/download.html
    - Sysinternals Suite https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx
    - Regshot http://sourceforge.net/projects/regshot/
    - Notepad++ http://notepad-plus-plus.org/
    - regfsnotify - https://github.com/mgoffin/malwarecookbook/tree/master/9/3
    - Visual Studio Express 2008: https://go.microsoft.com/?linkid=7729279
    - PDFStreamDumper https://zeltser.com/pdf-stream-dumper-malicious-file-analysis/
    - OfficeMalScanner http://www.reconstructer.org/code.html
    - Wireshark https://www.wireshark.org/

7. Create desktop shortcuts for the tools. From Sysinternals, you’ll want shortcuts for Process Monitor and Process Explorer.
8. Take a snapshot and name it “Clean with Tools”, or something that makes sense to you.

Configuring Virtual Networking:

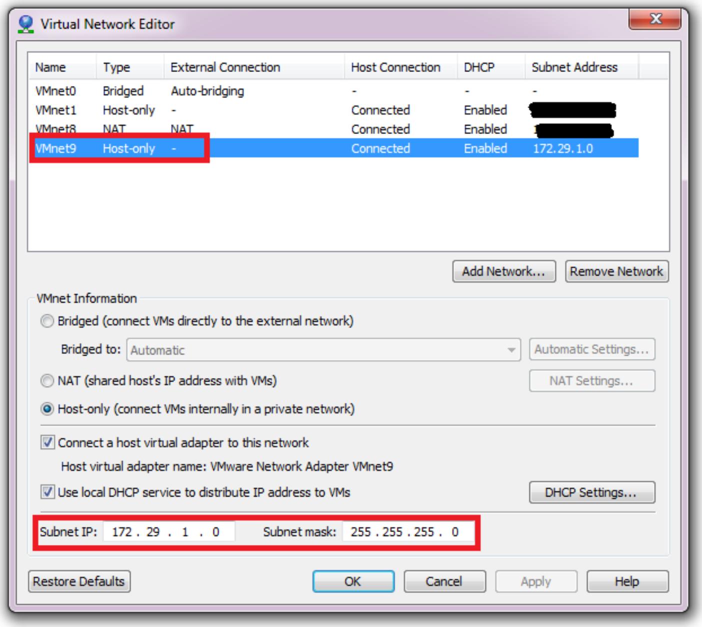
We are now going to link our VMs together into an isolated VMNet. None of the VMs will be able to access the internet, but they will be able to communicate with each other. What we are building will look something like the graphic below:


REMnux
Server for malware interaction, Linux based analysis
Win7x64
Static, Dynamic, and Behavioral Analysis
Isolated VMNet
Host

Building the Network

In VMWare Workstation Home -

1. Edit → Virtual Network Editor
2. Add Network, name it VMNet9. It should default to host only.
3. Select vmnet9 from the upper list
4. Set subnet IP in the lower portion of the window to 172.29.1.0 (click apply)



Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Auto-bridging	-	-	-
VMnet1	Host-only	-	Connected	Enabled	[redacted]
VMnet8	NAT	NAT	Connected	Enabled	[redacted]
VMnet9	Host-only	-	Connected	Enabled	172.29.1.0

VMnet Information

○ Bridged (connect VMs directly to the external network)

Bridged to: Automatic

○ NAT (shared host's IP address with VMs)

● Host-only (connect VMs internally in a private network)

☑ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet9

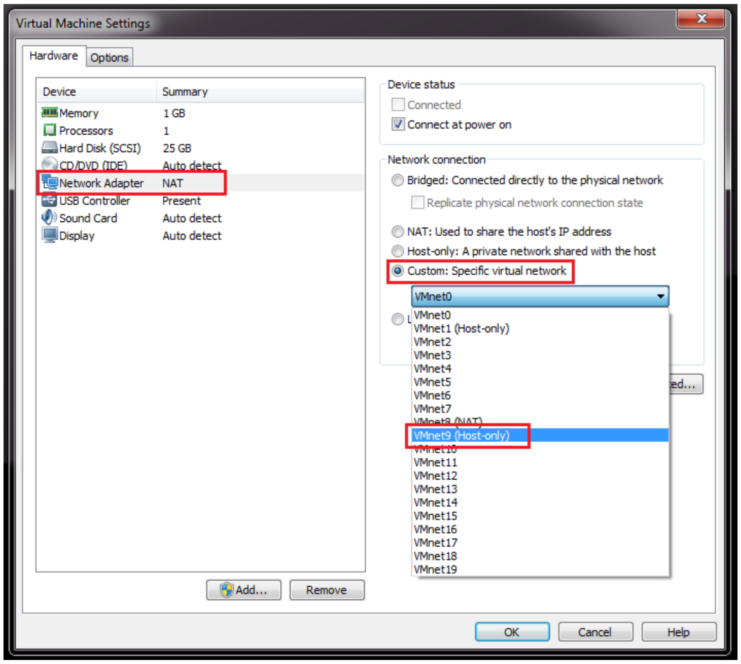
☑ Use local DHCP service to distribute IP address to VMs

Subnet IP: 172 . 29 . 1 . 0   Subnet mask: 255 . 255 . 255 . 0

Configuring REMnux Guest Virtual Networking:

In REMnux Tab of VMware Workstation

1. Rt. Click tab → Settings → Network Adapter → Custom → vmnet9 → click ok



Virtual Machine Settings

Device	Summary
Memory	1 GB
Processors	1
Hard Disk (SCSI)	25 GB
CD/DVD (IDE)	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Device status

□ Connected

☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

○ NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

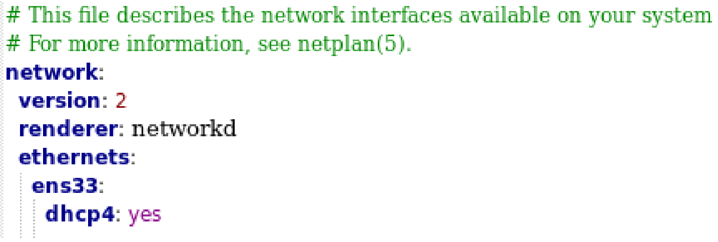
● Custom: Specific virtual network

VMnet0

Within the guest Machine, configure the OS to use ip 172.29.1.10, nm 255.255.255.0

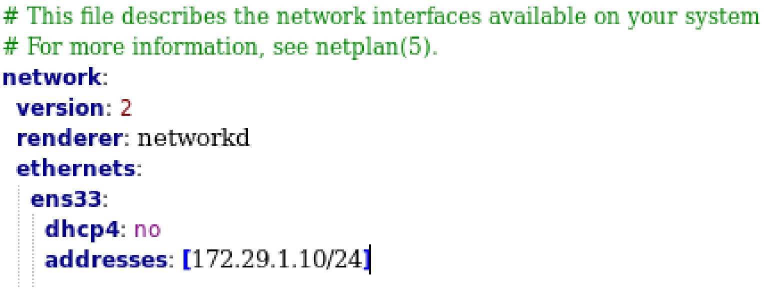
1. Backup the config file:
   ```
   >sudo cp /etc/netplan/01-netcfg.yaml /etc/netplan/01-netcfg.bak
   ```
2. Open `/etc/netplan/01-netcfg.yaml` as admin
   ```
   >sudo scite /etc/netplan/01-netcfg.yaml
   ```

Change:



```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:
      dhcp4: yes
```

To:



```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:
      dhcp4: no
      addresses: [172.29.1.10/24]
```

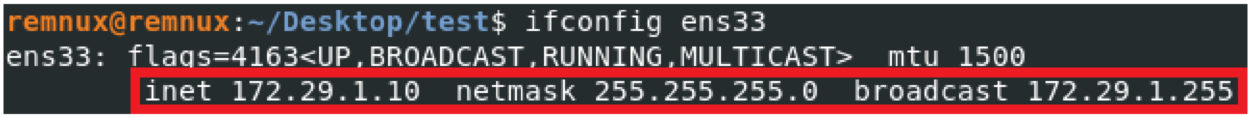
3. Apply the changes by executing the following on the command line:

```
> sudo netplan apply
```

4. Confirm you've correctly configured the network by executing the following on the command line:

```
> ifconfig ens33
```

The output should look something like this:



```
remnux@remnux:~/Desktop/test$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.29.1.10  netmask 255.255.255.0  broadcast 172.29.1.255
```

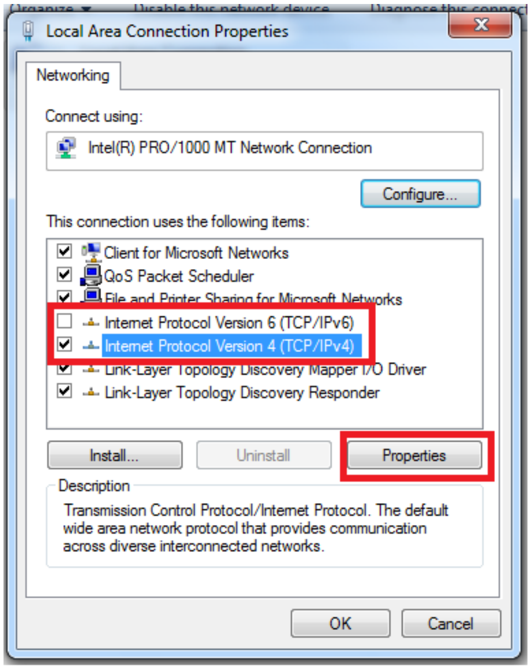
Configuring Windows Guest Virtual Networking:

In Win7_Toaster Tab of VMware Workstation-

1. Rt. Click tab → Settings → Network Adapter → Custom → vmnet9 → click ok

Within the guest OS

1. Start → Search → “Network and Sharing Center”
2. Left Menu, “Change Adaptor Settings”
3. Right click “Local Area Connection” → Properties
4. Uncheck “TCP/IPv6”
5. Check “TCP/IPv4”, click properties
6. Set these Values:
    a. IP Address: 172.29.1.20
    b. Subnet Mask: 255.255.255.0
    c. Default gateway: 172.29.1.10
    d. Preferred DNS Server 172.29.1.10
7. Click OK, close
8. Choose “Home Network” when prompted, click cancel when asked about sharing



Local Area Connection Properties

Networking

Connect using:

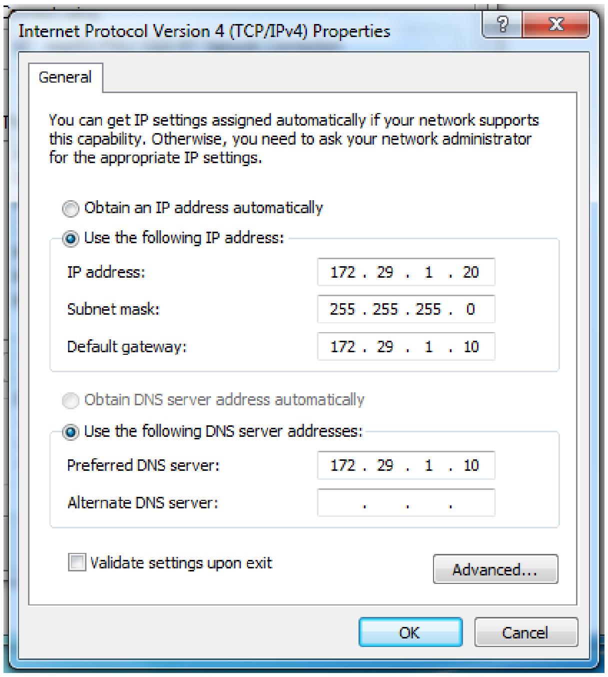
Intel(R) PRO/1000 MT Network Connection

This connection uses the following items:

☑ Client for Microsoft Networks
☑ QoS Packet Scheduler
☑ File and Printer Sharing for Microsoft Networks
☐ Internet Protocol Version 6 (TCP/IPv6)
☑ Internet Protocol Version 4 (TCP/IPv4)
☑ Link-Layer Topology Discovery Mapper I/O Driver
☑ Link-Layer Topology Discovery Responder

Description

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



Internet Protocol Version 4 (TCP/IPv4) Properties

General

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: 172 . 29 . 1 . 20
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 172 . 29 . 1 . 10

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: 172 . 29 . 1 . 10
Alternate DNS server: . . .

☐ Validate settings upon exit

Configuring Inetsim

Now we have all of our virtual machines on their own isolated network and we have the windows machines thinking that the remnux machine is their default gateway and DNS server. We're missing one piece though - Remnux doesn't actually know how to be those things. Fortunately there's a program called Inetsim that knows how to behave like a server, we just have to configure it.

Just in case you mess something up, make a copy of /etc/inetsim/inetsim.conf

```
> sudo cp /etc/inetsim/inetsim.conf /etc/inetsim/inetsim.bak
```

Open /etc/inetsim/inetsim.conf as admin:

```
> sudo scite /etc/inetsim/inetsim.conf
```

Make the following changes:

1. Find the line:

   ```
   #start_service dns
   ```

   And uncomment it so it looks like:

   ```
   start_service dns
   ```

2. Find the line:

   ```
   #service_bind_address 10.10.10.1
   ```

   Change it to read:

   ```
   service_bind_address 172.29.1.10
   ```

3. Find the line:

   ```
   #dns_default_ip 10.10.10.1
   ```

   Change it to read:

   ```
   dns_default_ip 172.29.1.10
   ```

Save and close the config file.

Testing your Configuration:

From Remnux

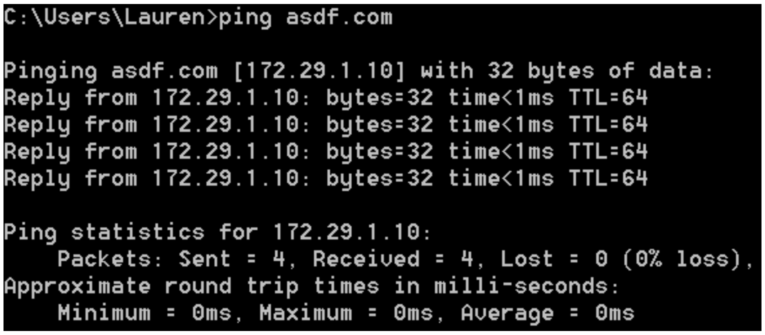
- Launch inetsim by simply typing “inetsim” into the terminal
- Open Wireshark and start capture

From Windows

- ping asdf.com

If everything is working properly you will see:

- ping responses from ip 172.29.1.10

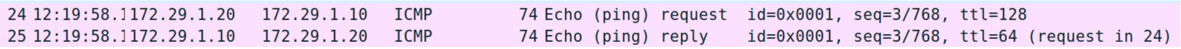


```
C:\Users\Lauren>ping asdf.com

Pinging asdf.com [172.29.1.10] with 32 bytes of data:
Reply from 172.29.1.10: bytes=32 time<1ms TTL=64
Reply from 172.29.1.10: bytes=32 time<1ms TTL=64
Reply from 172.29.1.10: bytes=32 time<1ms TTL=64
Reply from 172.29.1.10: bytes=32 time<1ms TTL=64

Ping statistics for 172.29.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

- In Wireshark, you should see a series of ping requests and ping replies, among other traffic



```
24 12:19:58.1172.29.1.20    172.29.1.10    ICMP         74 Echo (ping) request  id=0x0001, seq=3/768, ttl=128
25 12:19:58.1172.29.1.10    172.29.1.20    ICMP         74 Echo (ping) reply    id=0x0001, seq=3/768, ttl=64 (request in 24)
```

If everything is working properly, **snapshot** each VM. If things are not working properly, it’s time to troubleshoot. Review the instructions and double check your configurations. Feel free to email us for help.

Back Up Your Host:

In this class you will learn protocols that, if followed, will prevent you from detonating malware on your host. Nonetheless, mistakes happen; especially when we're learning. Consequently, I strongly recommend that you back your host up before coming to class.

How you choose to back your host up is up to you, but you should be able to restore your backup completely independent of your operating system. Additionally, your backup should not be connected to your host while you're performing malware analysis. Windows restore points are not sufficient.

Clonezilla and Norton Ghost are two examples of products that will take backups that meet these requirements.