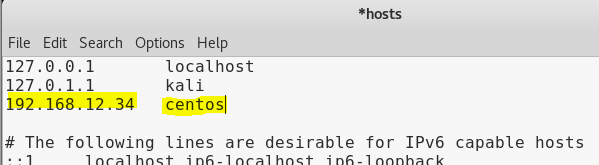
# SSH Extras--SCP and SSH Tunnels

Ubuntu note: This lab was originally written for student who had two VMs, CentOS and Kali. If you are using Ubuntu, there is only one change. CentOS server has an SSH server installed by default, but Ubuntu desktop does not. If you did the previous lab, Basic SSH Security Lab-Win-Ubuntu, you have already installed SSH server or sshd. Other than that, everything else is identical. Wherever the lab says to use the CentOS VM, use the Ubuntu VM instead.

One way to make typing easier for this lab is to add an entry in the /etc/hosts files for the IP addresses we are using. For example, we can configure the Kali VM so that all we have to do is type centos instead of the IP address for the CentOS VM. If the address of the CentOS VM is 192.168.12.34, add the following line to /etc/hosts on the Kali VM so that the typing is now “centos” versus “192.168.12.34.”   


Also, a host file entry on the CentOS VM with kali and the IP address of the Kali VM may be handy. The DNS client always looks in the hosts file first, so it will see the centos in the hosts file and use that address rather than trying to find it with a DNS query.

If you want to adjust the hosts file on a Windows machine, you can find it in C:\Windows\System32\drivers\etc\hosts. It works the same way that the hosts file works in Linux; Windows checks the hosts file before trying to find the address with a DNS query.

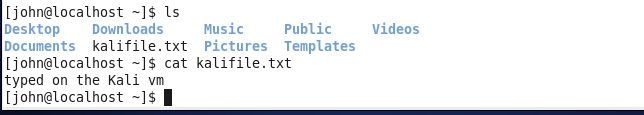
## Secure Copy (scp)

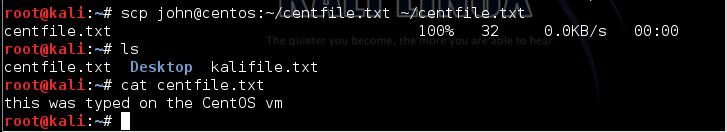
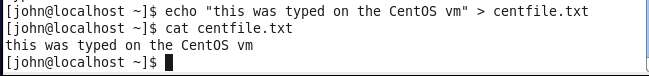
Once you have SSH running, you can also copy files back and forth securely using secure copy (scp). It’s quite easy, and doesn’t have the security problems that FTP does. The format is very similar to plain old copy (cp).  
cp [source file] [destination file]

The difference is that you specify the computer you are copying to or from using the same header that you use for SSH, [user]@[ip address or computer name]:. Examples are john@centos: or john@192.168.12.34:. Note the colon between john@centos and the file name!

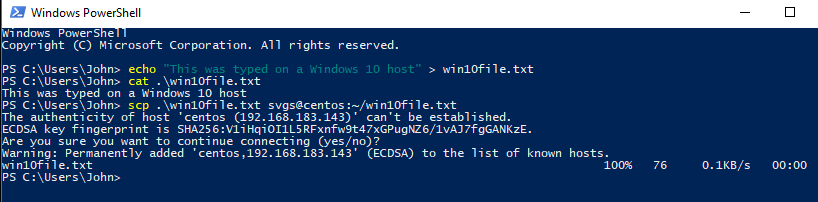
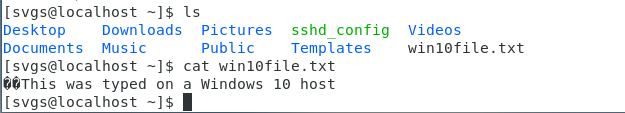
So, if I want to copy a file (called file, for example) from my local computer to a remote computer, I might use:  
scp ~/file user@remotecomputer:file

If I want to copy a file from the remote computer to my local home directory, I might use:  
scp user@remotecomputer:file ~/file  
  
Let’s make a file on Kali and then use scp to copy it to CentOS. 

The file is indeed on the CentOS VM. 

Now, let’s create a file on the CentOS VM, and then use scp on the Kali VM to fetch it. Since the source file is on the CentOS VM, we use john@centos: before the source file name.

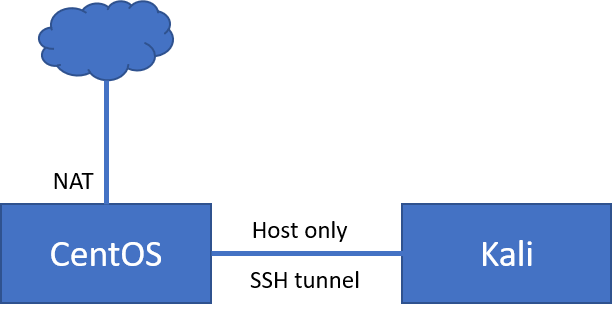
## SCP on Windows

Hurray! As of the April 2018 update to Windows 10 (version 1803) OpenSSH is now installed on Windows by default. It is the same code that Linux uses, so it works the same way.  
  


# Browsing through an SSH tunnel

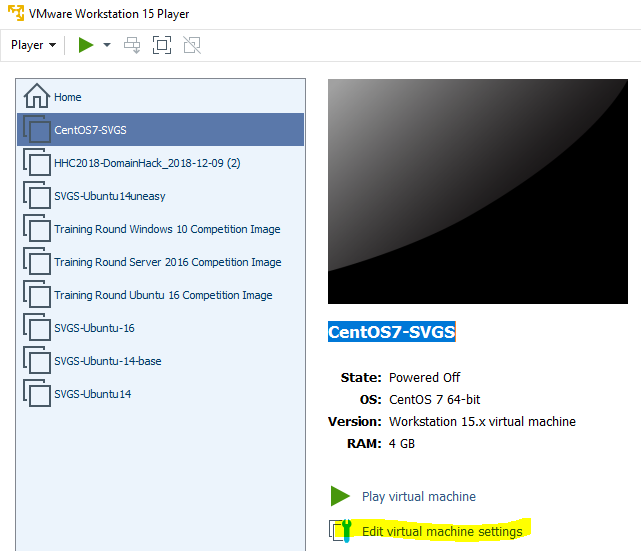
## Lab Setup

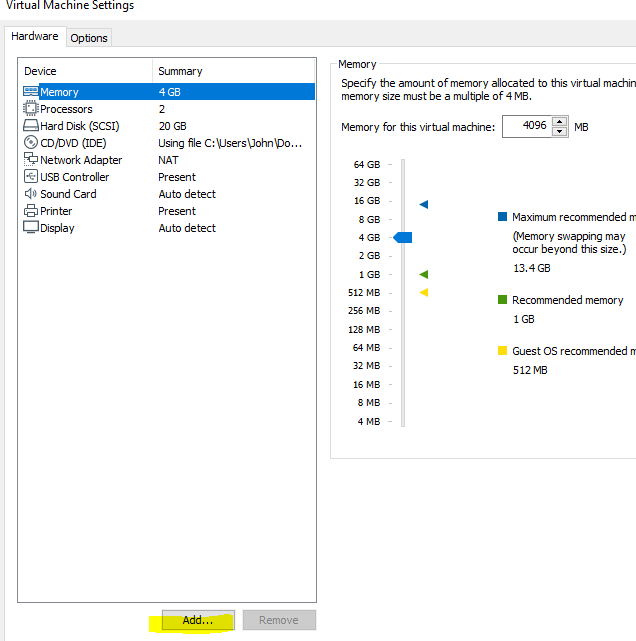
This lab needs an SSH client on one host or VM, and an SSH server (sshd) with Internet access on another host or VM. We will continue with our use of sshd on the CentOS VM and an SSH client on the Kali VM. To make it obvious that our traffic is indeed transiting an SSH tunnel, we will put the Kali VM on the Host-only network and add a second network interface on the CentOS VM that is on the Hot-only network. The tunnel will be from the Kali VM to the CentOS VM on the Host-only network, and then the CentOS VM will forward the traffic to the Internet on its NAT network interface.

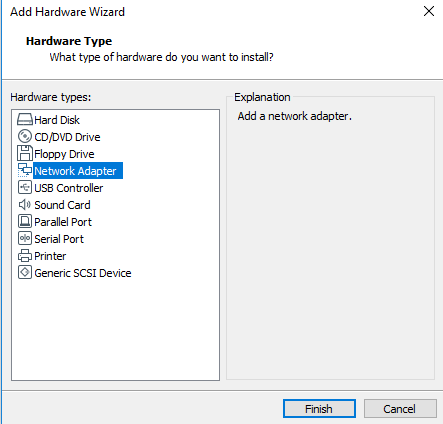
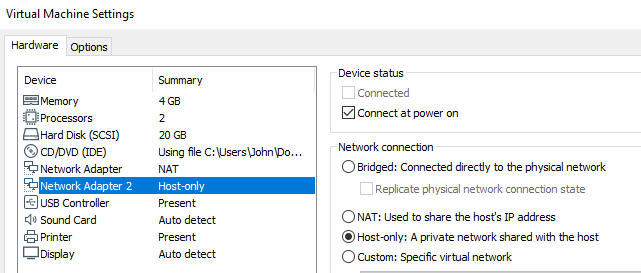


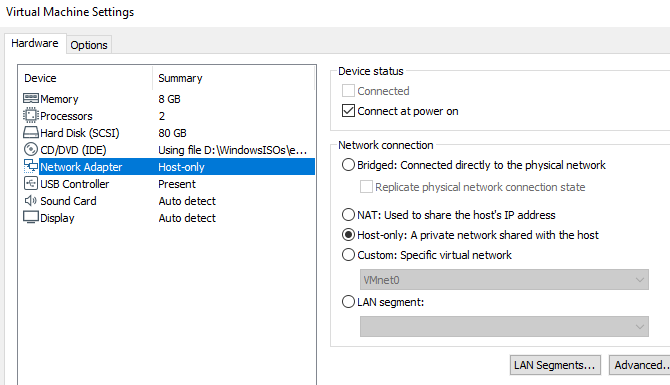
We will need to pay attention to the IP addresses, or it will get confusing.

### Configure VM networking

Add a second network interface to the CentOS VM by editing the VM settings. It is easiest to do this with the VM shut down.  


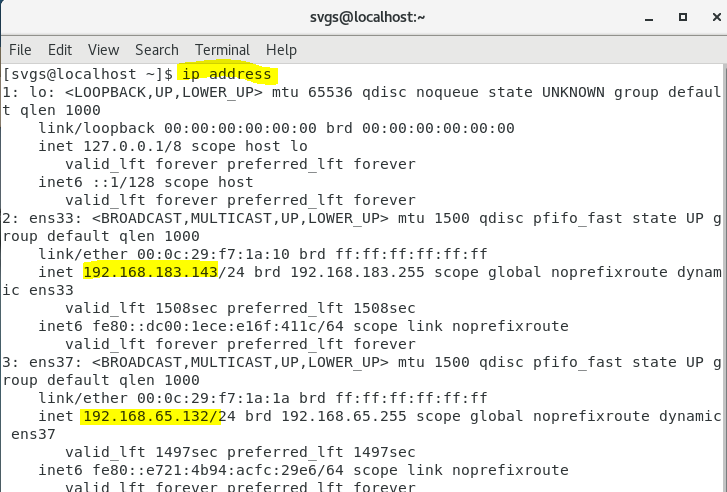
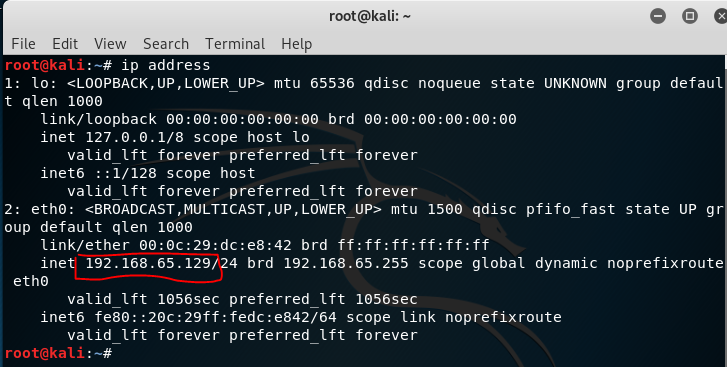
Select Add… at the bottom of the Virtual Machine Settings dialog box.  


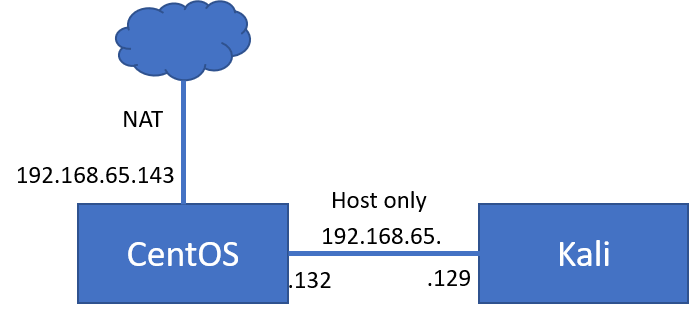
Then add a second network adapter and configure it to be on the Host-only network.  
  


On your Kali VM, change its network setting so that the adapter is on the Host-only network.  


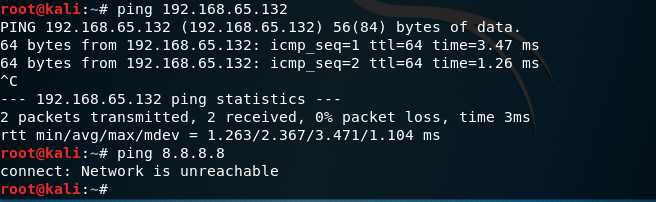
### Start the VMs and check the IP addresses

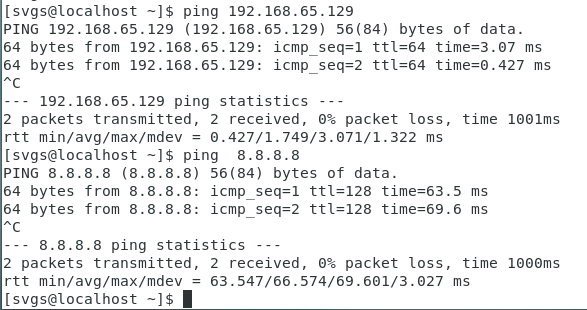
Power up the two VMs and use the ip address command on both the CentOS and the Kali VMs. The CentOS VM should have two addresses, and the Kali should have one.

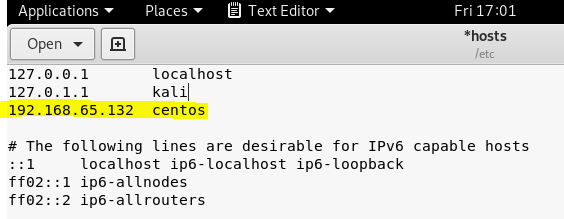
Then create a diagram showing the addresses and networks. Your addresses will be different than those shown below.  


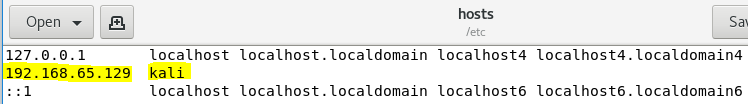
### Verify Connectivity

Verify that the Kali VM can ping the CentOS VM, but not the Internet.  


Verify that the CentOS VM can ping both the Kali VM and the Internet.  


### Update /etc/hosts

If you put entries in the /etc/hosts files so you can use centos and kali instead of IP addresses, remember to update them. This is /etc/hosts on the Kali VM.  


This is /etc/hosts on the CentOS VM. Remember, your addresses will be different.  


## SSH Tunnels

SSH tunnels come in three general flavors, local, dynamic, and remote. A local tunnel uses -L in the SSH command and allows one local port to be tunneled to a single remote host. For example, this command  
ssh -L 8000:www.somesite.com:80 svgs@centos  
takes any traffic received on local port 8000, puts it in an SSH tunnel to centos, and tells centos to forward the traffic to [www.somesite.com](http://www.somesite.com), port 80. You send the traffic you want to be tunneled to localhost:8000 or 127.0.0.1:8000. You can pick any port number for the local port (8000 in this case) if the port is not in use and you remember to send your data to that port. The port number at the end (80 in this case) must be the port that [www.somesite.com](http://www.somesite.com) listens on. Unfortunately, web sites have become complicated enough that local tunnels no longer work well for web browsing.

Local SSH tunnels do work well for other protocols, however. Windows file sharing over Server Message Block (SMB) protocol works well through SSH tunnels. There is an exercise later in the course that makes use of this.

## Dynamic SSH tunnels

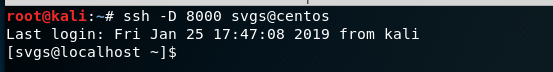
A dynamic SSH tunnel uses the SOCKSv5 Proxy protocol. SOCKSv5 allows the tunnel to send traffic to any host. It is not limited to a single host listed in the command line as in the SSH Local tunnel. It also has procedures for passing authentication traffic so that browsing works just as it would in a direct connection.

### Aside--web proxy servers

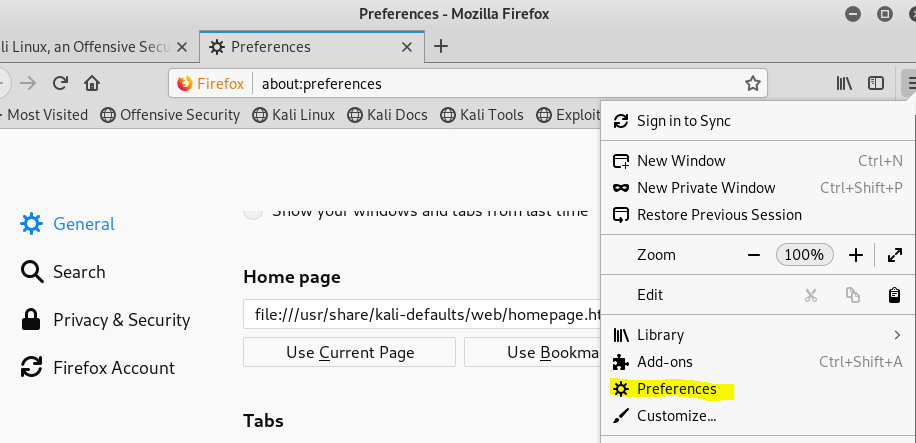
Wise organizations force all user web traffic to go through a proxy server, where attacks can be filtered, malware outbound Command and Control (CnC) traffic may be blocked, and all traffic can be logged. Nowadays, this function is often incorporated into the organization’s Next Generation Firewall (NGFW). In the past, the proxy was a separate appliance and the browser had to be told to send its traffic to the proxy server instead of the default gateway. Most browsers include settings that will allow us to direct the browser traffic to the proxy, or SSH tunnel in our case, instead of to the default gateway.

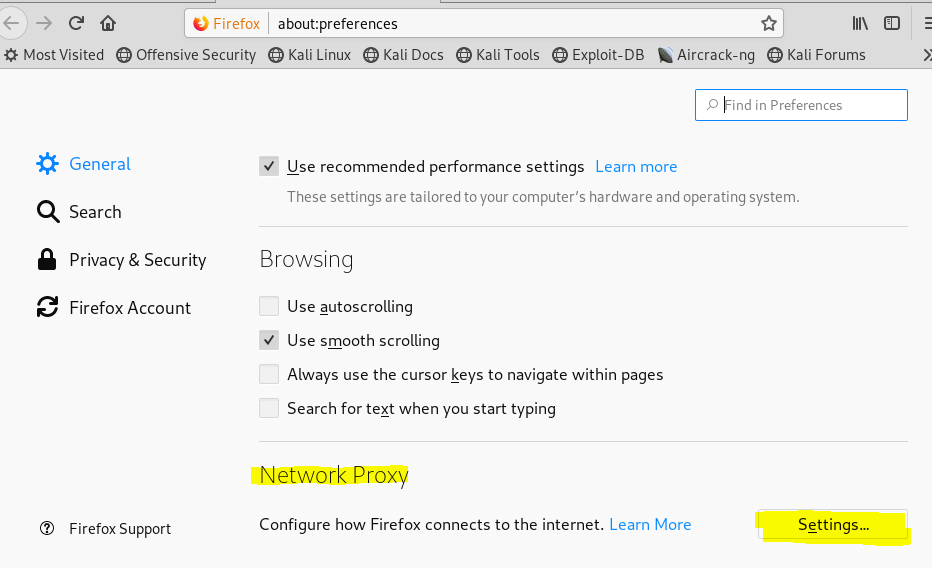
### Create the tunnel

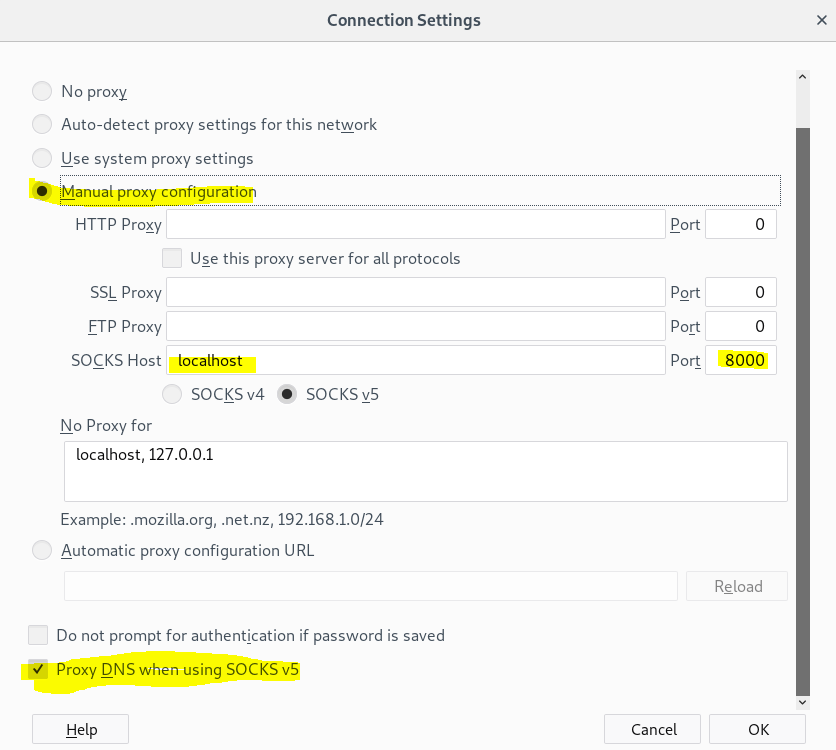
On the Kali VM, open an SSH connection to the CentOS VM, using the dynamic tunnel option  
-D [port number]. The port number can be anything we choose if the port is not already in use. It is also good to choose a port number above 1024, since Linux requires root privileges to use ports below 1024.

Here’s the SSH connection, using local port 8000 (our choice, can be anything).  
ssh –D 8000 svgs@centos  
 

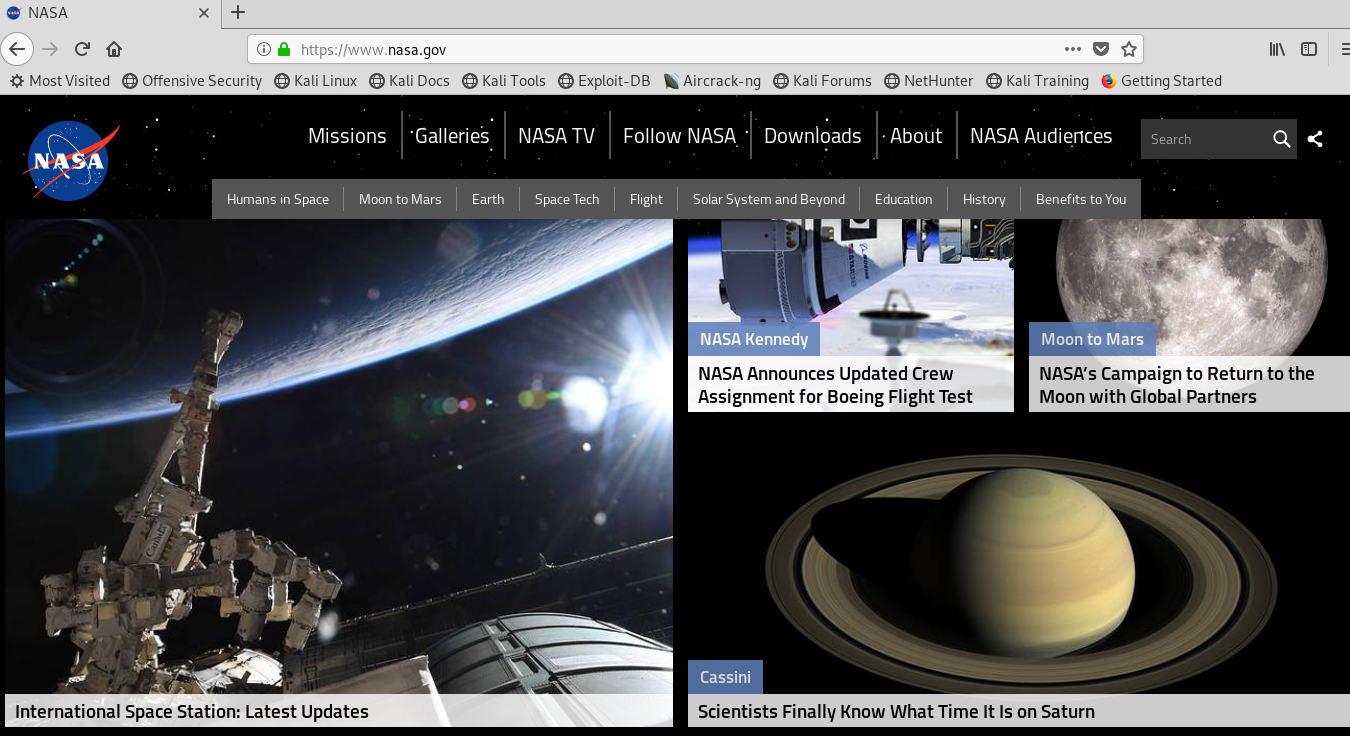
### Configure Proxy Settings on the Kali Firefox browser

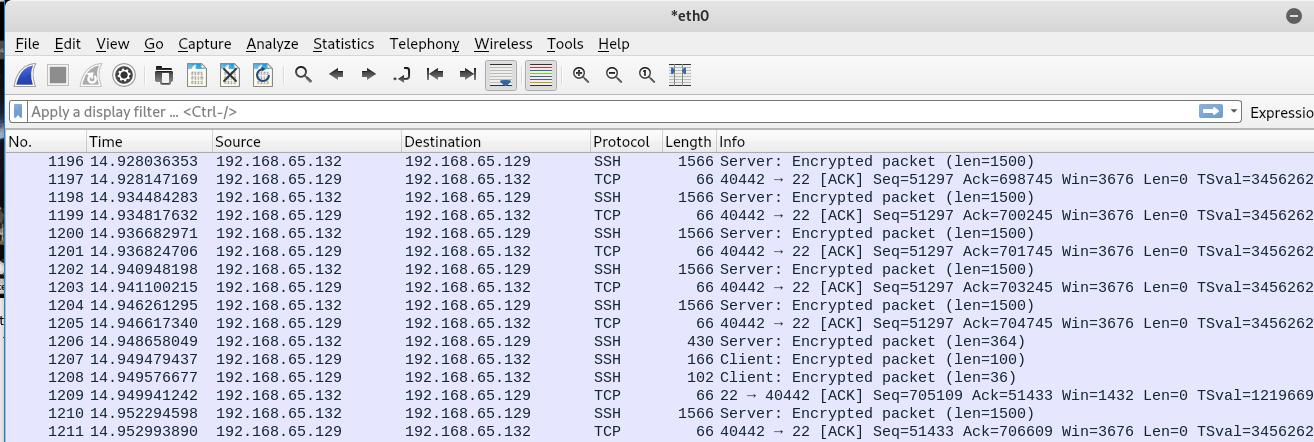
On Kali, select Preferences and then General.  


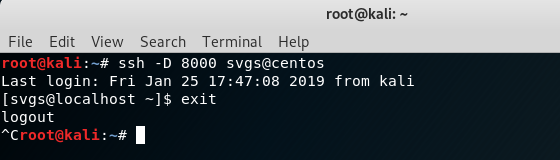
Scroll to the very bottom, and select Network Proxy Settings.  


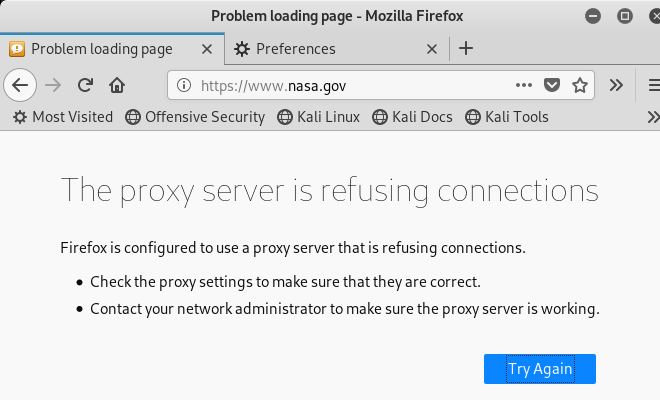
Manually configure the proxy to use SOCKSv5. The address in the SOCKS Host field can be either localhost or 127.0.0.1. The port number should match whatever we picked when we created the tunnel. Also select “Proxy DNS when using SOCKSv5”.  


Note: People doing web application penetration tests change proxies so often they install browser plugins to allow them to change proxy information from the main browser window.

With the dynamic tunnel configured we can browse normally. The browser will send its requests through the encrypted SSH tunnel to the CentOS machine. The CentOS VM will forward them to the Internet, receive the answers, and send them back to the Kali browser through the tunnel. 

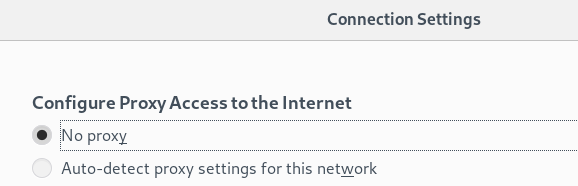
If you want to verify that our browsing is really going through the tunnel, you can run Wireshark or tcpdump to examine the traffic. All the traffic is SSH traffic to the CentOS VM. There is no HTTP or HTTPS traffic visible since it is inside the encrypted SSH tunnel.  


Another way to verify that the traffic is using the SSH tunnel is to disconnect the SSH session.  


The proxy server (SSH dynamic tunnel using SOCKSv5) is no longer running, so the browser can no longer connect to the Internet.  


## Return the VMs to their normal configuration

On the CentOS VM, delete the second network adapter (Host-only) we added.

On the Kali VM, change the network adapter back to the NAT network, and set the Firefox browser back to “No proxy”.  


## Note

Network connections in VMware Workstation Player work like hubs instead of switches; a VM will see all traffic on its network, not just traffic that is addressed to the VM. If we used our original configuration with the CentOS and Kali VMs both on the NAT network, and just one network adapter on the CentOS VM, the Wireshark capture would have been confusing. We would see the SSH connection, like the one above, but we would also see the traffic from the CentOS VM to and from the Internet. It would not have been so obvious that the browser traffic was using only the SSH tunnel.