COMP8006: Assignment 2

Login Monitor Blocker

Kevin Lo, A00952922

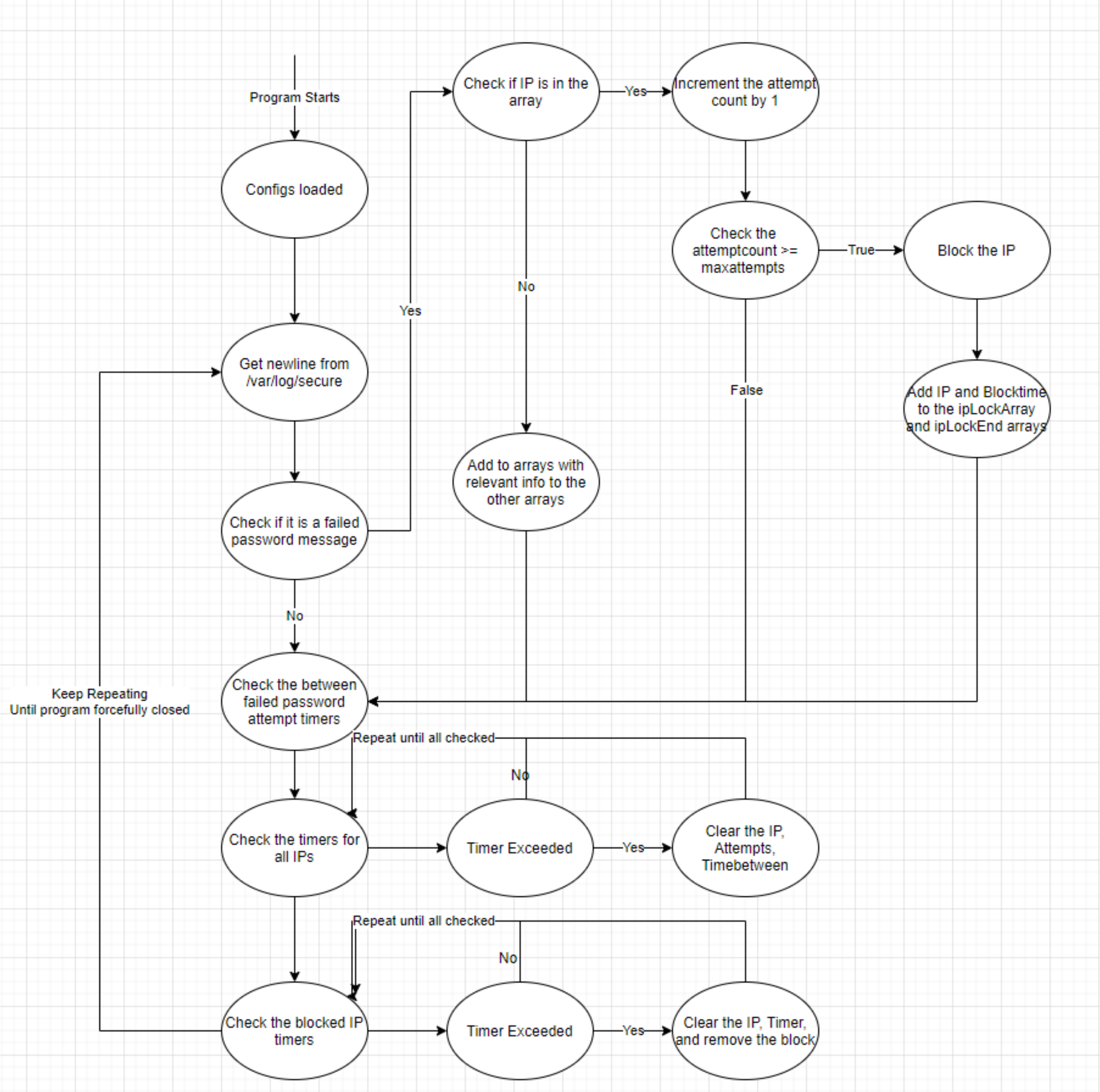
2-22-2021

Table of Contents

1 Introduction

This assignment was to design a program that can detect failed password attempts to remotely access a computer through such as SSH or Telnet. Upon reaching the user specified maximum attempts allowed within the user specified timeframe between attempts, this program will automatically create an iptables rule to block the IP which made the attempts for a specified amount of time.

1.1 Finite State Machine


Program Starts
Configs loaded
Get newline from /var/log/secure
Check if it is a failed password message
Yes
No
Check if IP is in the array
Yes
No
Increment the attempt count by 1
Check the attemptcount >= maxattempts
True
False
Block the IP
Add IP and Blocktime to the ipLockArray and ipLockEnd arrays
Add to arrays with relevant info to the other arrays
Check the between failed password attempt timers
Keep Repeating Until program forcefully closed
Repeat until all checked
No
Check the timers for all IPs
Timer Exceeded
Yes
Clear the IP, Attempts, Timebetween
Repeat until all checked
No
Check the blocked IP timers
Timer Exceeded
Yes
Clear the IP, Timer, and remove the block

1.2 Pseudocode

```
import re #for regular expression

read entire /var/log/secure file until current
    do nothing

ipArray = []
ipAttemptCount = []
ipTimeBetween = []
ipLockArray = []
ipLockEnd = []

#repeat the following 3 steps until program is forcefully canceled

constantly read from the /var/log/secure file #waiting for new lines
#look for line containing failed password from IP
if (Failed password for user from x port)
    ip = re.findall( r'[0-9]+(?:\.[0-9]+){3}', line )
    write the line containing the IP to file
    check if the ip is in the table
        #in table
        if (currentTime <= ipTimeBetween[])
            ipAttemptCount[]++
            if (count == maximumAttempts)
                add the iptables rule
                ipLockArray.add()
                ipLockEnd.add()
            else
            ipAttemptCount[] = 1
        #not in table
        else
            ipArray.add()
            ipAttemptCount.add(1)
            ipTimeBetween.add()
#timerchecks
#check the "timeout"
for i in range(len(ipTimeBetween))
    if currentTime > ipTimeLimit[i]
        ipAddr.pop(i)
        ipAttemptCount.pop(i)
        ipTimeLimit.pop(i)
#blocktimercheck
for i in range(len(ipLockArray))
    if currentTime > ipLockEnd[i]
        #execute rule to unblock ip
        ipLockEnd.pop(i)
        ipLockArray.pop(i)
```

2 How to Use

The following section teaches how to use the Python script with the configuration file to monitor the /var/log/secure file for failed remote access attempts.

2.1 Physical Requirement

Only requires one machine to have the Python script to run on that you want secured.

2.2 Operating System

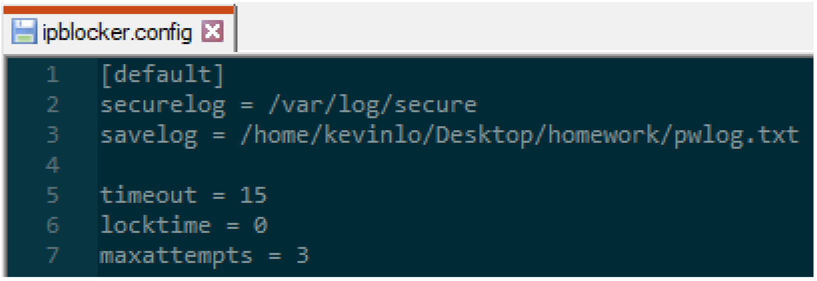
The machine should be a variant of Linux, in my case it is Fedora which uses /var/log/secure to save the log files. Other distros may use /var/log/secure so be sure to configure the settings to match your distro.

2.3 Setting up the System

rsyslog needs to be installed to allow the logging of system events, specifically failed password events.

2.4 Configuring the ipblocker.config File

The config file contains settings on how the ipblocker program will function.



```
[default]
securelog = /var/log/secure
savelog = /home/kevinlo/Desktop/homework/pwlog.txt

timeout = 15
locktime = 0
maxattempts = 3
```

securelog: Location of your /var/log/secure file
savelog: Location to save all instances of failed remote access password attempts ssh/telnet
timeout: The maximum time allowed in seconds between failed password attempts, if this time expires, the attempts are set back to 0
locktime: The amount of time duration in seconds to block a connection who has exceeded their max attempts. If this is set to 0, the IP will be blocked permanently
maxattempts: The maximum number of attempts permitted before an IP is blocked.

2.5 Notable Configurations

Setting the timeout to a higher number may help avoid the brute forcing of the password however this may inconvenience users potentially blocking users from accessing the server and needing the admin to unblock their IP.

For most setups having locktime set to 0 to block IPs permanently will be preferred for most users as they are expected to know the password

3 Testing Details

I use three different computers for the testing

-Server Computer (Fedora) (IP: 192.168.1.250)

-SSH Computer (Rasbian) (IP: 192.168.1.252)

-Telnet Computer (Windows) (IP: 192.168.1.253)

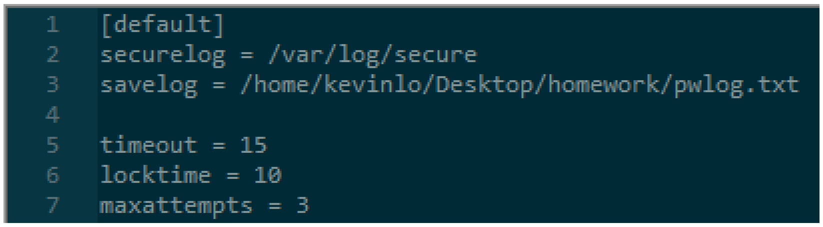
The SSH computer is used to SSH into the server computer, the Telnet computer is used to Telnet into the server computer.

4 Tests

Test #	Test Description	Tool Used	Expected Result	Pass/Fail
1	Do **maxattempts** amount failed password attempts using SSH	SSH	The IP address is blocked for the **blocktime** amount of time, and then unblocked	Pass. Detailed results attached.
2	Do **maxattempts** amount failed password attempts using Telnet	Telnet	The IP address is blocked for the **blocktime** specified amount of time, and then unblocked	Pass. Detailed results attached.

4.1 Test Results

Configuration Used for Testing

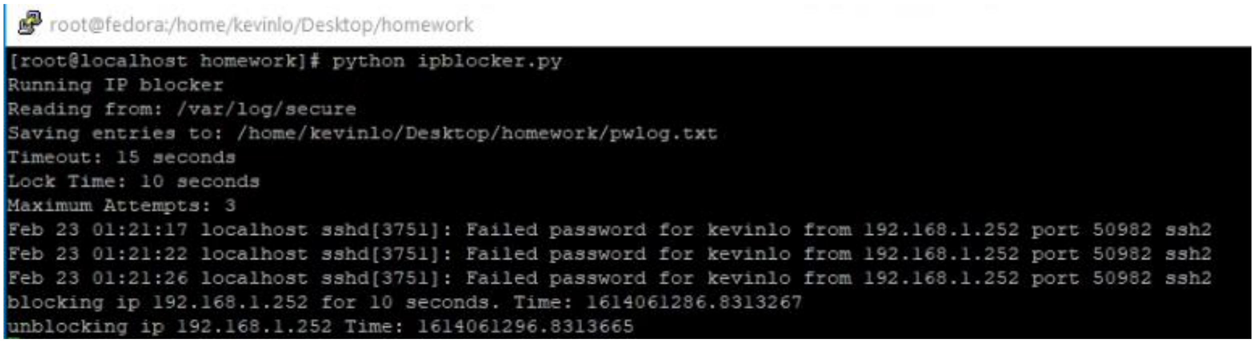


```
[default]
securelog = /var/log/secure
savelog = /home/kevinlo/Desktop/homework/pwlog.txt

timeout = 15
locktime = 10
maxattempts = 3
```

This is the configuration used to test the functionality of blocking and unblocking IPs after a specified amount of time.

4.1.1 Test 1 – Failed SSH Password 3 Attempts

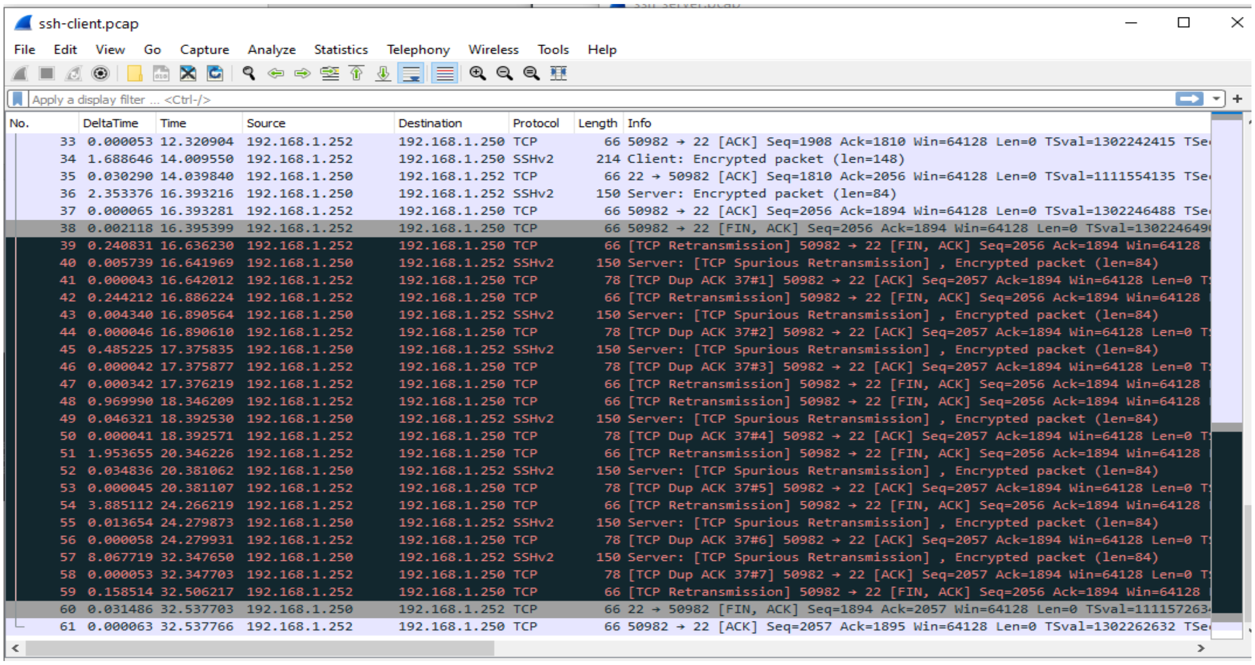
Ipblocker.py output



root@fedora:/home/kevinlo/Desktop/homework

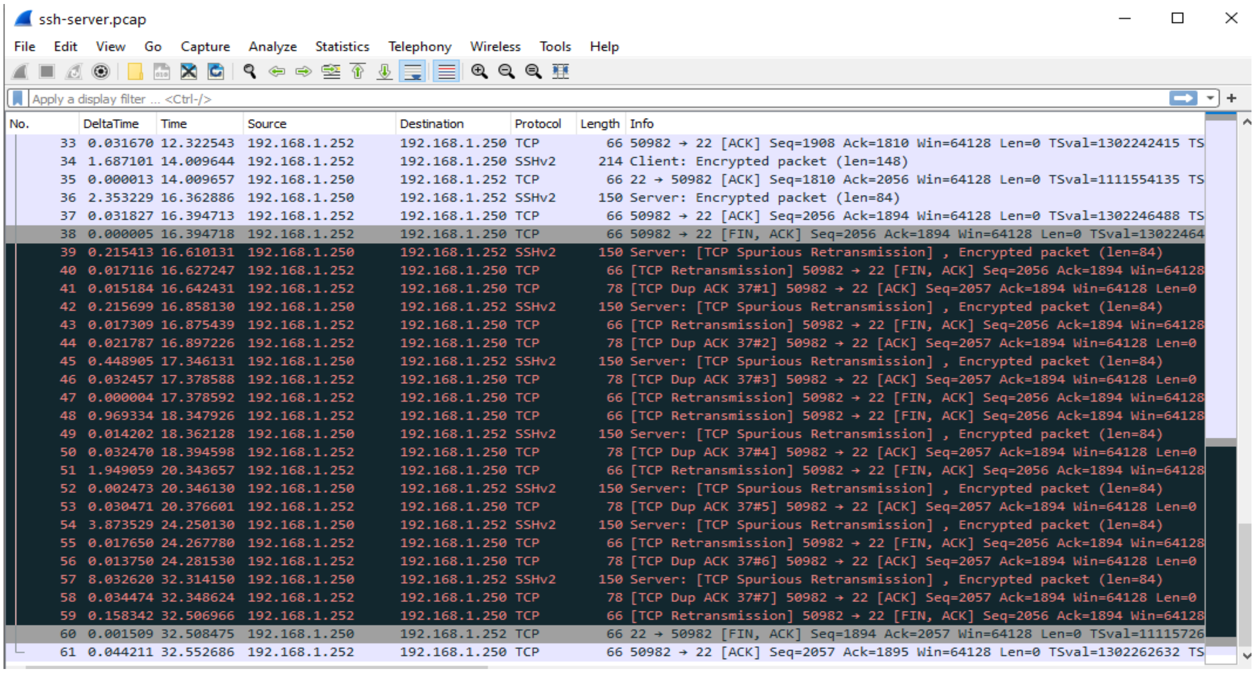
```
[root@localhost homework]# python ipblocker.py
Running IP blocker
Reading from: /var/log/secure
Saving entries to: /home/kevinlo/Desktop/homework/pwlog.txt
Timeout: 15 seconds
Lock Time: 10 seconds
Maximum Attempts: 3
Feb 23 01:21:17 localhost sshd[3751]: Failed password for kevinlo from 192.168.1.252 port 50982 ssh2
Feb 23 01:21:22 localhost sshd[3751]: Failed password for kevinlo from 192.168.1.252 port 50982 ssh2
Feb 23 01:21:26 localhost sshd[3751]: Failed password for kevinlo from 192.168.1.252 port 50982 ssh2
blocking ip 192.168.1.252 for 10 seconds. Time: 1614061286.8313267
unblocking ip 192.168.1.252 Time: 1614061296.8313665
```

Client packet capture



No.	DeltaTime	Time	Source	Destination	Protocol	Length	Info
33	0.000053	12.320904	192.168.1.252	192.168.1.250	TCP	66	50982 → 22 [ACK] Seq=1908 Ack=1810 Win=64128 Len=0 TSval=1302242415 TSe
34	1.688646	14.009550	192.168.1.252	192.168.1.250	SSHv2	214	Client: Encrypted packet (len=148)
35	0.030290	14.039840	192.168.1.250	192.168.1.252	TCP	66	22 → 50982 [ACK] Seq=1810 Ack=2056 Win=64128 Len=0 TSval=1111554135 TSe
36	2.353376	16.393216	192.168.1.250	192.168.1.252	SSHv2	150	Server: Encrypted packet (len=84)
37	0.000065	16.393281	192.168.1.252	192.168.1.250	TCP	66	50982 → 22 [ACK] Seq=2056 Ack=1894 Win=64128 Len=0 TSval=1302246488 TSe
38	0.002118	16.395399	192.168.1.252	192.168.1.250	TCP	66	50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128 Len=0 TSval=130224649
39	0.240831	16.636230	192.168.1.252	192.168.1.250	TCP	66	[TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
40	0.005739	16.641969	192.168.1.250	192.168.1.252	SSHv2	150	Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
41	0.000043	16.642012	192.168.1.252	192.168.1.250	TCP	78	[TCP Dup ACK 37#1] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0 T
42	0.244212	16.886224	192.168.1.252	192.168.1.250	TCP	66	[TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
43	0.004340	16.890564	192.168.1.250	192.168.1.252	SSHv2	150	Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
44	0.000046	16.890610	192.168.1.252	192.168.1.250	TCP	78	[TCP Dup ACK 37#2] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0 T
45	0.485225	17.375835	192.168.1.250	192.168.1.252	SSHv2	150	Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
46	0.000042	17.375877	192.168.1.252	192.168.1.250	TCP	78	[TCP Dup ACK 37#3] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0 T
47	0.000342	17.376219	192.168.1.252	192.168.1.250	TCP	66	[TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
48	0.969990	18.346209	192.168.1.252	192.168.1.250	TCP	66	[TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
49	0.046321	18.392530	192.168.1.250	192.168.1.252	SSHv2	150	Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
50	0.000041	18.392571	192.168.1.252	192.168.1.250	TCP	78	[TCP Dup ACK 37#4] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0 T
51	1.953655	20.346226	192.168.1.252	192.168.1.250	TCP	66	[TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
52	0.034836	20.381062	192.168.1.250	192.168.1.252	SSHv2	150	Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
53	0.000045	20.381107	192.168.1.252	192.168.1.250	TCP	78	[TCP Dup ACK 37#5] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0 T
54	3.885112	24.266219	192.168.1.252	192.168.1.250	TCP	66	[TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
55	0.013654	24.279873	192.168.1.250	192.168.1.252	SSHv2	150	Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
56	0.000058	24.279931	192.168.1.252	192.168.1.250	TCP	78	[TCP Dup ACK 37#6] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0 T
57	8.067719	32.347650	192.168.1.250	192.168.1.252	SSHv2	150	Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
58	0.000053	32.347703	192.168.1.252	192.168.1.250	TCP	78	[TCP Dup ACK 37#7] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0 T
59	0.158514	32.506217	192.168.1.252	192.168.1.250	TCP	66	[TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
60	0.031486	32.537703	192.168.1.250	192.168.1.252	TCP	66	22 → 50982 [FIN, ACK] Seq=1894 Ack=2057 Win=64128 Len=0 TSval=111157263
61	0.000063	32.537766	192.168.1.252	192.168.1.250	TCP	66	50982 → 22 [ACK] Seq=2057 Ack=1895 Win=64128 Len=0 TSval=1302262632 TSe

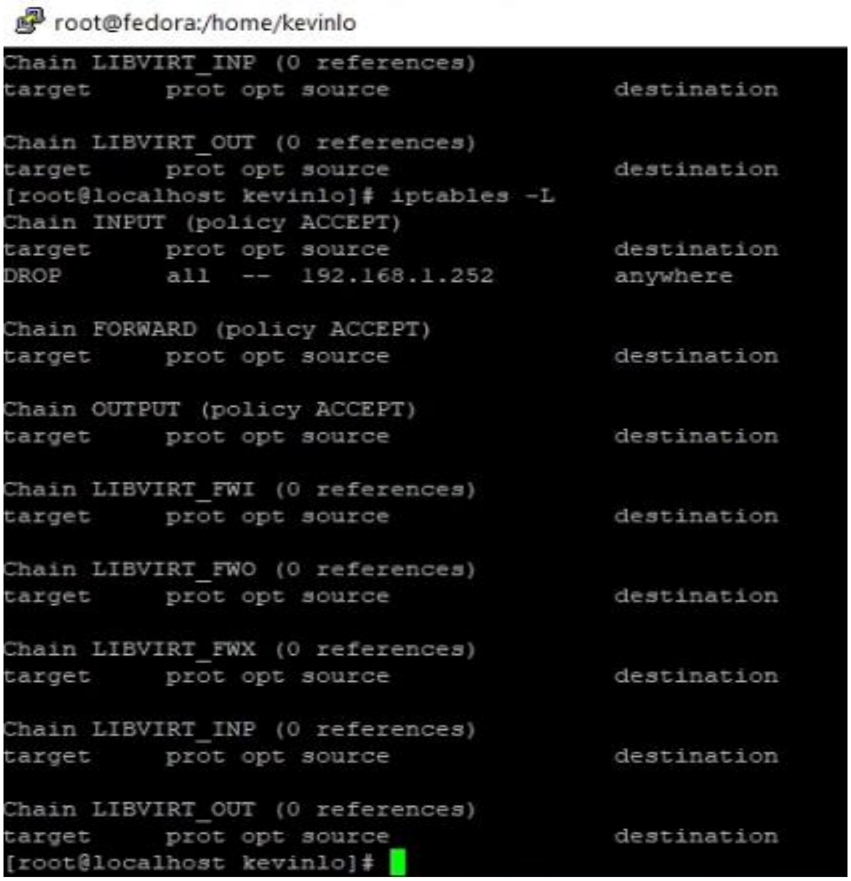
Server Packet capture



ssh-server.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  DeltaTime Time      Source        Destination   Protocol Length Info
  33 0.031670 12.322543 192.168.1.252 192.168.1.250 TCP      66 50982 → 22 [ACK] Seq=1908 Ack=1810 Win=64128 Len=0 TSval=1302242415 TS
  34 1.687101 14.009644 192.168.1.252 192.168.1.250 SSHv2   214 Client: Encrypted packet (len=148)
  35 0.000013 14.009657 192.168.1.250 192.168.1.252 TCP      66 22 → 50982 [ACK] Seq=1810 Ack=2056 Win=64128 Len=0 TSval=1111554135 TS
  36 2.353229 16.362886 192.168.1.250 192.168.1.252 SSHv2   150 Server: Encrypted packet (len=84)
  37 0.031827 16.394713 192.168.1.252 192.168.1.250 TCP      66 50982 → 22 [ACK] Seq=2056 Ack=1894 Win=64128 Len=0 TSval=1302246488 TS
  38 0.000005 16.394718 192.168.1.252 192.168.1.250 TCP      66 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128 Len=0 TSval=13022464
  39 0.215413 16.610131 192.168.1.250 192.168.1.252 SSHv2   150 Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
  40 0.017116 16.627247 192.168.1.252 192.168.1.250 TCP      66 [TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
  41 0.015184 16.642431 192.168.1.252 192.168.1.250 TCP      78 [TCP Dup ACK 37#1] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0
  42 0.215699 16.858130 192.168.1.250 192.168.1.252 SSHv2   150 Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
  43 0.017309 16.875439 192.168.1.252 192.168.1.250 TCP      66 [TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
  44 0.021787 16.897226 192.168.1.252 192.168.1.250 TCP      78 [TCP Dup ACK 37#2] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0
  45 0.448905 17.346131 192.168.1.250 192.168.1.252 SSHv2   150 Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
  46 0.032457 17.378588 192.168.1.252 192.168.1.250 TCP      78 [TCP Dup ACK 37#3] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0
  47 0.000004 17.378592 192.168.1.252 192.168.1.250 TCP      66 [TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
  48 0.969334 18.347926 192.168.1.252 192.168.1.250 TCP      66 [TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
  49 0.014202 18.362128 192.168.1.250 192.168.1.252 SSHv2   150 Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
  50 0.032470 18.394598 192.168.1.252 192.168.1.250 TCP      78 [TCP Dup ACK 37#4] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0
  51 1.949059 20.343657 192.168.1.252 192.168.1.250 TCP      66 [TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
  52 0.002473 20.346130 192.168.1.250 192.168.1.252 SSHv2   150 Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
  53 0.030471 20.376601 192.168.1.252 192.168.1.250 TCP      78 [TCP Dup ACK 37#5] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0
  54 3.873529 24.250130 192.168.1.250 192.168.1.252 SSHv2   150 Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
  55 0.017650 24.267780 192.168.1.252 192.168.1.250 TCP      66 [TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
  56 0.013750 24.281530 192.168.1.252 192.168.1.250 TCP      78 [TCP Dup ACK 37#6] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0
  57 8.032620 32.314150 192.168.1.250 192.168.1.252 SSHv2   150 Server: [TCP Spurious Retransmission] , Encrypted packet (len=84)
  58 0.034474 32.348624 192.168.1.252 192.168.1.250 TCP      78 [TCP Dup ACK 37#7] 50982 → 22 [ACK] Seq=2057 Ack=1894 Win=64128 Len=0
  59 0.158342 32.506966 192.168.1.252 192.168.1.250 TCP      66 [TCP Retransmission] 50982 → 22 [FIN, ACK] Seq=2056 Ack=1894 Win=64128
  60 0.001509 32.508475 192.168.1.250 192.168.1.252 TCP      66 22 → 50982 [FIN, ACK] Seq=1894 Ack=2057 Win=64128 Len=0 TSval=11115726
  61 0.044211 32.552686 192.168.1.252 192.168.1.250 TCP      66 50982 → 22 [ACK] Seq=2057 Ack=1895 Win=64128 Len=0 TSval=1302262632 TS
```

Iptables entry created by the ipblocker.py for the block duration

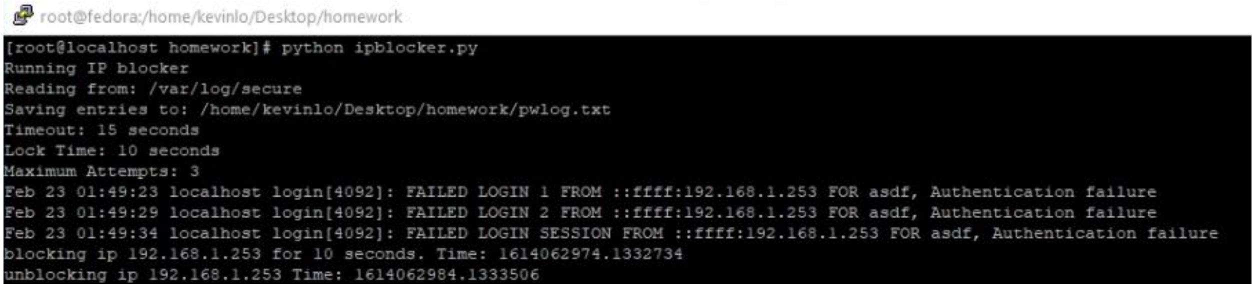


```
Chain LIBVIRT_INP (0 references)
target     prot opt source               destination

Chain LIBVIRT_OUT (0 references)
target     prot opt source               destination
[root@localhost kevinlo]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  192.168.1.252        anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain LIBVIRT_FWI (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWO (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWX (0 references)
target     prot opt source               destination

Chain LIBVIRT_INP (0 references)
target     prot opt source               destination

Chain LIBVIRT_OUT (0 references)
target     prot opt source               destination
[root@localhost kevinlo]#
```

This is the iptables during the time the IP is blocked for 10 seconds. The packet captures show that the computer that failed the password 3 times within the time limit was blocked successfully and unblocked successfully.

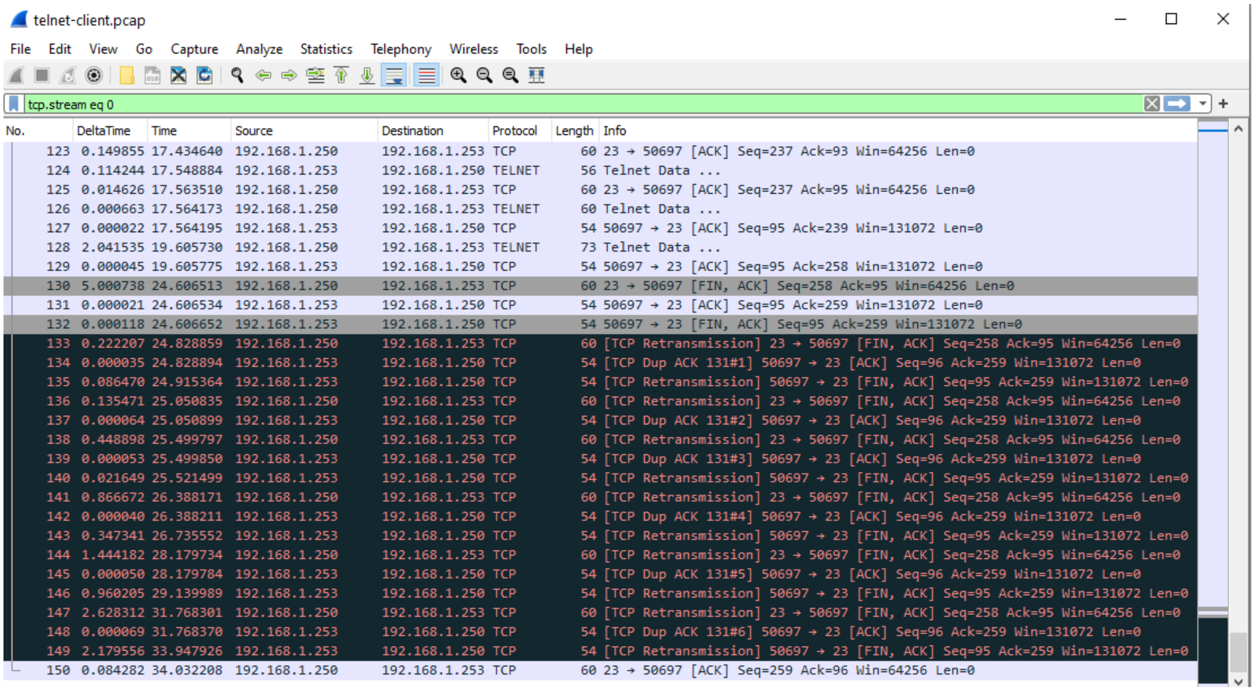
4.1.2 Test 2 – Failed Telnet Password 3 Attempts

Ipblocker.py output



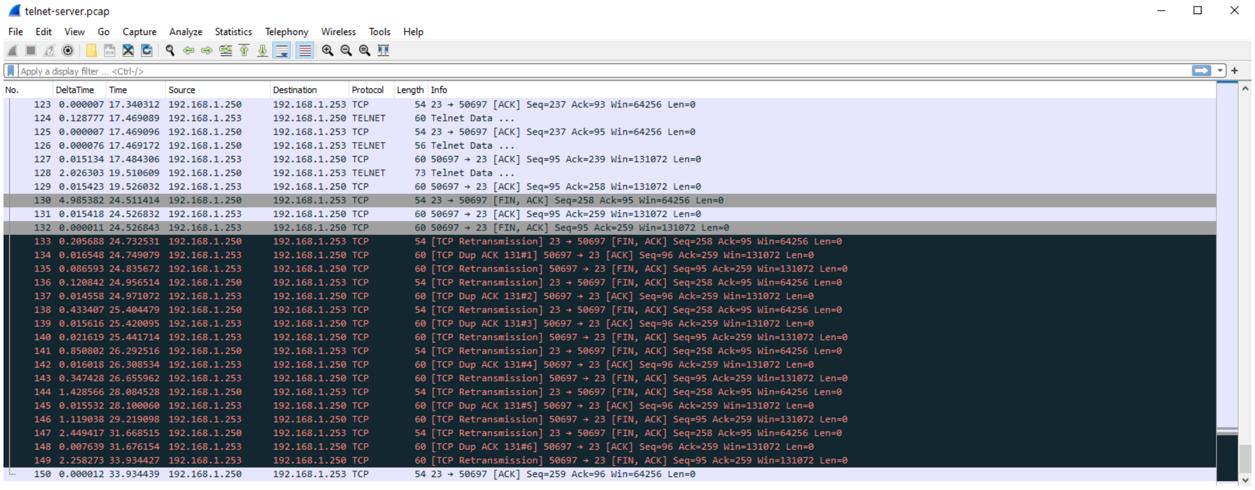
```
[root@localhost homework]# python ipblocker.py
Running IP blocker
Reading from: /var/log/secure
Saving entries to: /home/kevinlo/Desktop/homework/pwlog.txt
Timeout: 15 seconds
Lock Time: 10 seconds
Maximum Attempts: 3
Feb 23 01:49:23 localhost login[4092]: FAILED LOGIN 1 FROM ::ffff:192.168.1.253 FOR asdf, Authentication failure
Feb 23 01:49:29 localhost login[4092]: FAILED LOGIN 2 FROM ::ffff:192.168.1.253 FOR asdf, Authentication failure
Feb 23 01:49:34 localhost login[4092]: FAILED LOGIN SESSION FROM ::ffff:192.168.1.253 FOR asdf, Authentication failure
blocking ip 192.168.1.253 for 10 seconds. Time: 1614062974.1332734
unblocking ip 192.168.1.253 Time: 1614062984.1333506
```

Client packet capture



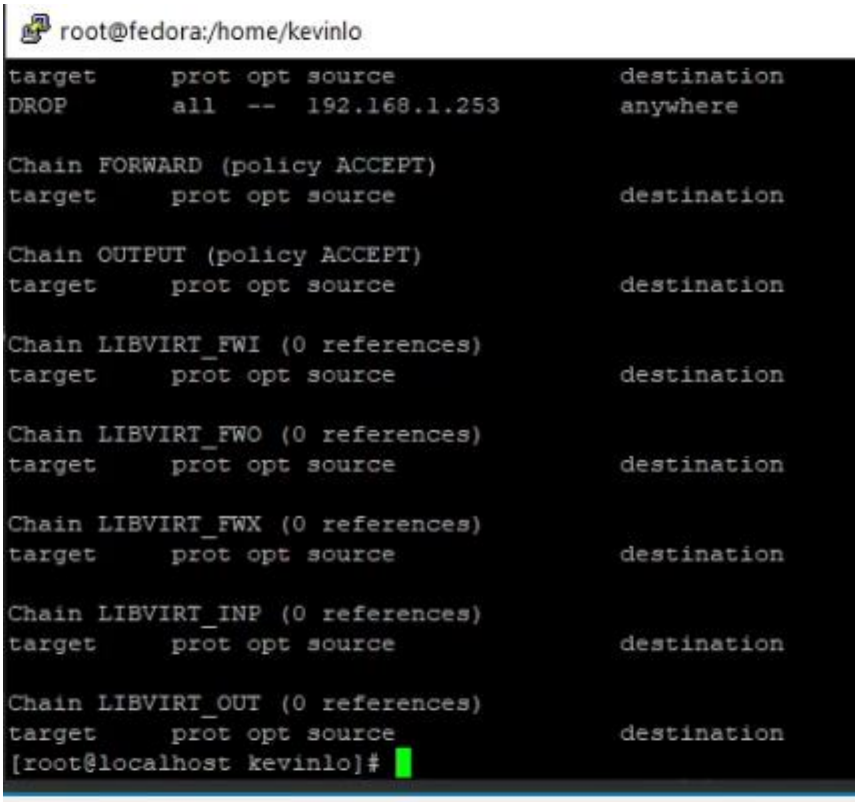
No.	DeltaTime	Time	Source	Destination	Protocol	Length	Info
123	0.149855	17.434640	192.168.1.250	192.168.1.253	TCP	60	23 → 50697 [ACK] Seq=237 Ack=93 Win=64256 Len=0
124	0.114244	17.548884	192.168.1.253	192.168.1.250	TELNET	56	Telnet Data ...
125	0.014626	17.563510	192.168.1.250	192.168.1.253	TCP	60	23 → 50697 [ACK] Seq=237 Ack=95 Win=64256 Len=0
126	0.000663	17.564173	192.168.1.250	192.168.1.253	TELNET	60	Telnet Data ...
127	0.000022	17.564195	192.168.1.253	192.168.1.250	TCP	54	50697 → 23 [ACK] Seq=95 Ack=239 Win=131072 Len=0
128	2.041535	19.605730	192.168.1.250	192.168.1.253	TELNET	73	Telnet Data ...
129	0.000045	19.605775	192.168.1.253	192.168.1.250	TCP	54	50697 → 23 [ACK] Seq=95 Ack=258 Win=131072 Len=0
130	5.000738	24.606513	192.168.1.250	192.168.1.253	TCP	60	23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
131	0.000021	24.606534	192.168.1.253	192.168.1.250	TCP	54	50697 → 23 [ACK] Seq=95 Ack=259 Win=131072 Len=0
132	0.000118	24.606652	192.168.1.253	192.168.1.250	TCP	54	50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
133	0.222207	24.828859	192.168.1.250	192.168.1.253	TCP	60	[TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
134	0.000035	24.828894	192.168.1.253	192.168.1.250	TCP	54	[TCP Dup ACK 131#1] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
135	0.086470	24.915364	192.168.1.253	192.168.1.250	TCP	54	[TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
136	0.135471	25.050835	192.168.1.250	192.168.1.253	TCP	60	[TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
137	0.000064	25.050899	192.168.1.253	192.168.1.250	TCP	54	[TCP Dup ACK 131#2] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
138	0.448898	25.499797	192.168.1.250	192.168.1.253	TCP	60	[TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
139	0.000053	25.499850	192.168.1.253	192.168.1.250	TCP	54	[TCP Dup ACK 131#3] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
140	0.021649	25.521499	192.168.1.253	192.168.1.250	TCP	54	[TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
141	0.866672	26.388171	192.168.1.250	192.168.1.253	TCP	60	[TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
142	0.000040	26.388211	192.168.1.253	192.168.1.250	TCP	54	[TCP Dup ACK 131#4] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
143	0.347341	26.735552	192.168.1.253	192.168.1.250	TCP	54	[TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
144	1.444182	28.179734	192.168.1.250	192.168.1.253	TCP	60	[TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
145	0.000050	28.179784	192.168.1.253	192.168.1.250	TCP	54	[TCP Dup ACK 131#5] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
146	0.960205	29.139989	192.168.1.253	192.168.1.250	TCP	54	[TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
147	2.628312	31.768301	192.168.1.250	192.168.1.253	TCP	60	[TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
148	0.000069	31.768370	192.168.1.253	192.168.1.250	TCP	54	[TCP Dup ACK 131#6] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
149	2.179556	33.947926	192.168.1.253	192.168.1.250	TCP	54	[TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
150	0.084282	34.032208	192.168.1.250	192.168.1.253	TCP	60	23 → 50697 [ACK] Seq=259 Ack=96 Win=64256 Len=0

Server packet capture


telnet-server.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   DeltaTime Time      Source        Destination   Protocol Length Info
  123 0.000007 17.340312 192.168.1.250 192.168.1.253 TCP      54 23 → 50697 [ACK] Seq=237 Ack=93 Win=64256 Len=0
  124 0.128777 17.469089 192.168.1.253 192.168.1.250 TELNET   60 Telnet Data ...
  125 0.000007 17.469096 192.168.1.250 192.168.1.253 TCP      54 23 → 50697 [ACK] Seq=237 Ack=95 Win=64256 Len=0
  126 0.000076 17.469172 192.168.1.250 192.168.1.253 TELNET   56 Telnet Data ...
  127 0.015134 17.484306 192.168.1.253 192.168.1.250 TCP      60 50697 → 23 [ACK] Seq=95 Ack=239 Win=131072 Len=0
  128 2.026303 19.510609 192.168.1.250 192.168.1.253 TELNET   73 Telnet Data ...
  129 0.015423 19.526032 192.168.1.253 192.168.1.250 TCP      60 50697 → 23 [ACK] Seq=95 Ack=258 Win=131072 Len=0
  130 4.985382 24.511414 192.168.1.250 192.168.1.253 TCP      54 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
  131 0.015418 24.526832 192.168.1.253 192.168.1.250 TCP      60 50697 → 23 [ACK] Seq=95 Ack=259 Win=131072 Len=0
  132 0.000011 24.526843 192.168.1.253 192.168.1.250 TCP      60 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
  133 0.205688 24.732531 192.168.1.250 192.168.1.253 TCP      54 [TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
  134 0.016548 24.749079 192.168.1.253 192.168.1.250 TCP      60 [TCP Dup ACK 131#1] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
  135 0.086593 24.835672 192.168.1.253 192.168.1.250 TCP      60 [TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
  136 0.120842 24.956514 192.168.1.250 192.168.1.253 TCP      54 [TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
  137 0.014558 24.971072 192.168.1.253 192.168.1.250 TCP      60 [TCP Dup ACK 131#2] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
  138 0.433407 25.404479 192.168.1.250 192.168.1.253 TCP      54 [TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
  139 0.015616 25.420095 192.168.1.253 192.168.1.250 TCP      60 [TCP Dup ACK 131#3] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
  140 0.021619 25.441714 192.168.1.253 192.168.1.250 TCP      60 [TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
  141 0.850802 26.292516 192.168.1.250 192.168.1.253 TCP      54 [TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
  142 0.016018 26.308534 192.168.1.253 192.168.1.250 TCP      60 [TCP Dup ACK 131#4] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
  143 0.347428 26.655962 192.168.1.253 192.168.1.250 TCP      60 [TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
  144 1.428566 28.084528 192.168.1.250 192.168.1.253 TCP      54 [TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
  145 0.015532 28.100060 192.168.1.253 192.168.1.250 TCP      60 [TCP Dup ACK 131#5] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
  146 1.119038 29.219098 192.168.1.253 192.168.1.250 TCP      60 [TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
  147 2.449417 31.668515 192.168.1.250 192.168.1.253 TCP      54 [TCP Retransmission] 23 → 50697 [FIN, ACK] Seq=258 Ack=95 Win=64256 Len=0
  148 0.007639 31.676154 192.168.1.253 192.168.1.250 TCP      60 [TCP Dup ACK 131#6] 50697 → 23 [ACK] Seq=96 Ack=259 Win=131072 Len=0
  149 2.258273 33.934427 192.168.1.253 192.168.1.250 TCP      60 [TCP Retransmission] 50697 → 23 [FIN, ACK] Seq=95 Ack=259 Win=131072 Len=0
  150 0.000012 33.934439 192.168.1.250 192.168.1.253 TCP      54 23 → 50697 [ACK] Seq=259 Ack=96 Win=64256 Len=0
```

Iptables entry created by ipblocker.py during the block duration



```
target     prot opt source               destination
DROP       all  --  192.168.1.253        anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain LIBVIRT_FWI (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWO (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWX (0 references)
target     prot opt source               destination

Chain LIBVIRT_INP (0 references)
target     prot opt source               destination

Chain LIBVIRT_OUT (0 references)
target     prot opt source               destination
[root@localhost kevinlo]#
```

This is the iptables during the time the IP is blocked for 10 seconds. The packet captures show that the computer that failed the password 3 times within the time limit was blocked successfully and unblocked successfully.

4.2 Verdict

2 out of 2 tests were successful.

5 Conclusion

This report is for my ipblocker.py Python program which monitors the /var/log/secure file for failed passwords for remote access and meets the requirements of automatically blocking the IP addresses who enter the wrong password too many times. The blocking of the IP address is done using iptables. This report demonstrates how to setup, test, and edit the configuration of my ipblocker program to suit the needs of the user.