COMP 8006
Assignment 3

Max Chung

A00743631

Set 6D

Table of Contents

Network Layout 3
Server Setup 4
Server Clean Up 6
Test Cases 8
Test Case 1 9
Test Case 2 13

Table of Figures

Figure 1 - The network layout 3
Figure 2 - Launching Terminal 4
Figure 3 - Making the script executable and running the script 4
Figure 4 - Specifying the blocking threshold and blocking duration 5
Figure 5 - Show all processes even ones without controlling terminals and the user running each process that contain the string “block-ip5.sh” 5
Figure 6 - block-ip5.sh listed in ps meaning the script is running 6
Figure 7 - Show all processes even ones without controlling terminals and the user running each process that contain the string “block-ip5.sh” 6
Figure 8 - block-ip5.sh listed in ps meaning the script is running 7
Figure 9 - Send SIGKILL (9) signal to process 13447 to end the process 7
Figure 10 - Show all processes even ones without controlling terminals and the user running each process that contain the string “block-ip5.sh” 7
Figure 11 - block-ip5.sh is no longer listed in ps meaning the script has terminated 8
Figure 12 - Login attempts blocked after four unsuccessful tries 9
Figure 13 - iptables rule and crontab event created to block offending IP address and to remove the iptables rule and the crontab rule itself after 30 minutes 10
Figure 14 - Banned IP address event created in /var/log/secure 11
Figure 15 - iptables rule and crontab event removed after the blocking duration elapsed 12
Figure 16 - Successful login removes history of the previous three failed login attempts 13
Figure 17 - No iptables blocking rules nor crontab events created by the script 14
Figure 18 - No banned IP address event created in /var/log/secure 15

Max Chung
A00743631
Set 6D

Network Layout

The layout of the network during the tests is as shown below where Fedora 1 and Fedora 2 are virtual machines running in Hyper-V and static IP addresses 192.168.0.50 and 192.168.0.51 are assigned respectively to the virtual machines. All computers in the network are capable of accessing the internet through the Cisco DPC3825 modem.

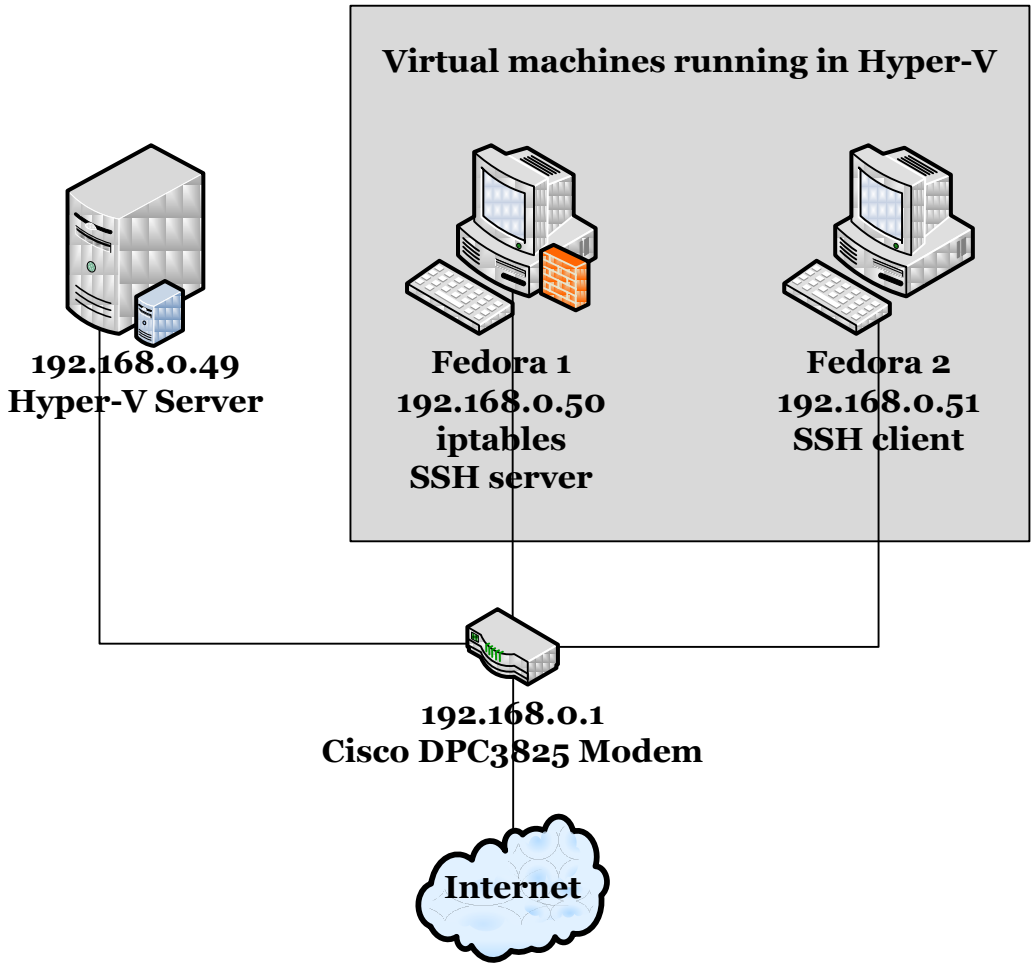

Figure 1 - The network layout

Server Setup

Before running any of the IP blocking tests, the server environment needs to be configured. The steps to configure the server are covered below.

Click on the Applications menu on the top left hand corner and select System Tools/Terminal

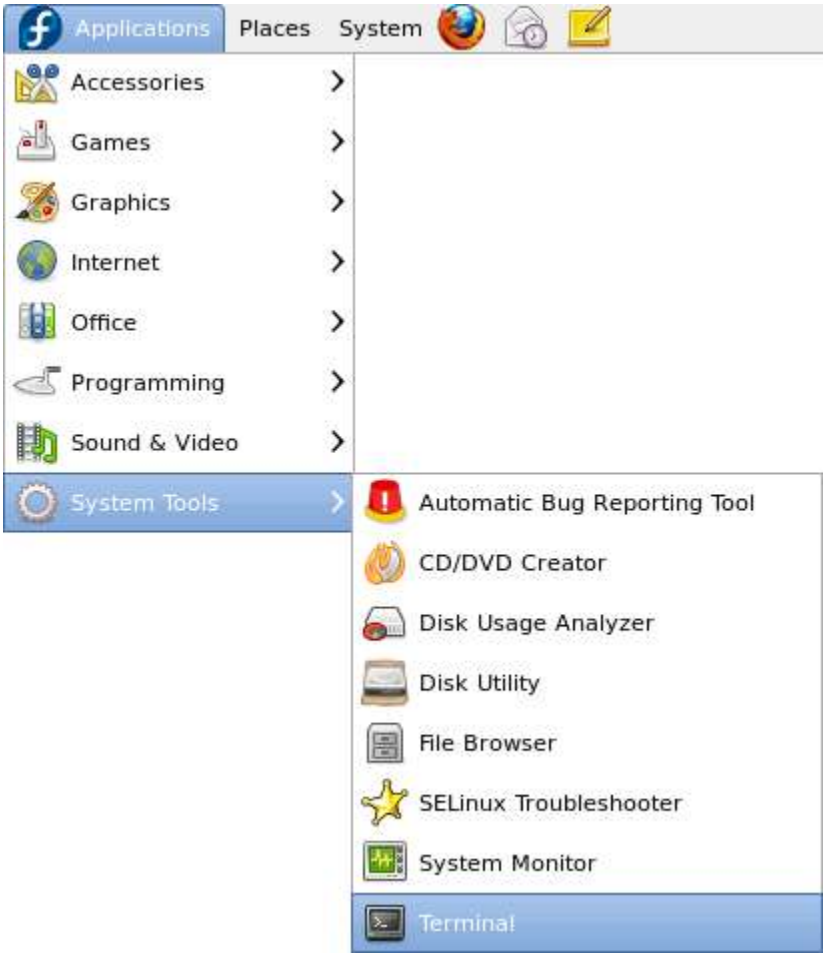

Figure 2 - Launching Terminal

Once the terminal window appears, type and execute the following commands:

```
su
<Your root password>
chmod +x ./block-ip5.sh
./block-ip5.sh
```

Figure 3 - Making the script executable and running the script

Specify 3 as the blocking threshold and 30 as the blocking duration in minutes.

The script will add an entry in crontab to run this script one minute later before quitting.

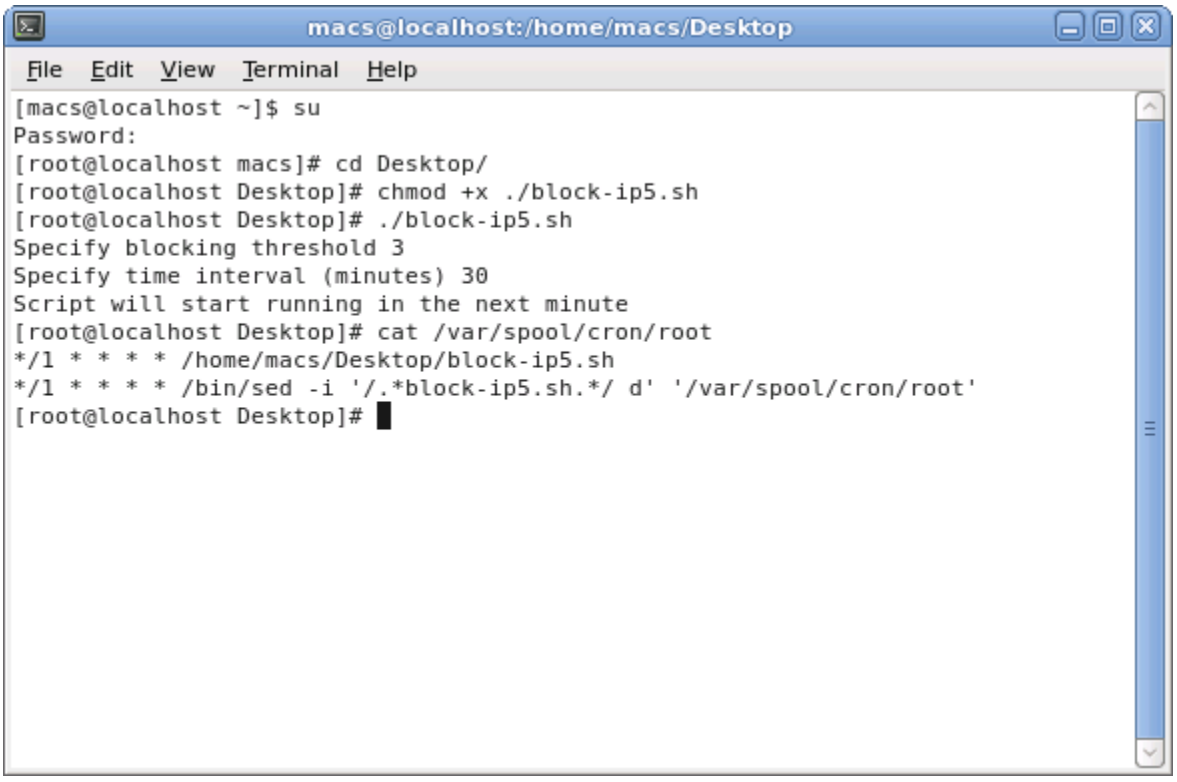

Figure 4 - Specifying the blocking threshold and blocking duration

Ensure that the block-ip5.sh script is running one minute later by executing the following command in terminal:

```
ps aux | grep block-ip5.sh
```

Figure 5 - Show all processes even ones without controlling terminals and the user running each process that contain the string "block-ip5.sh"

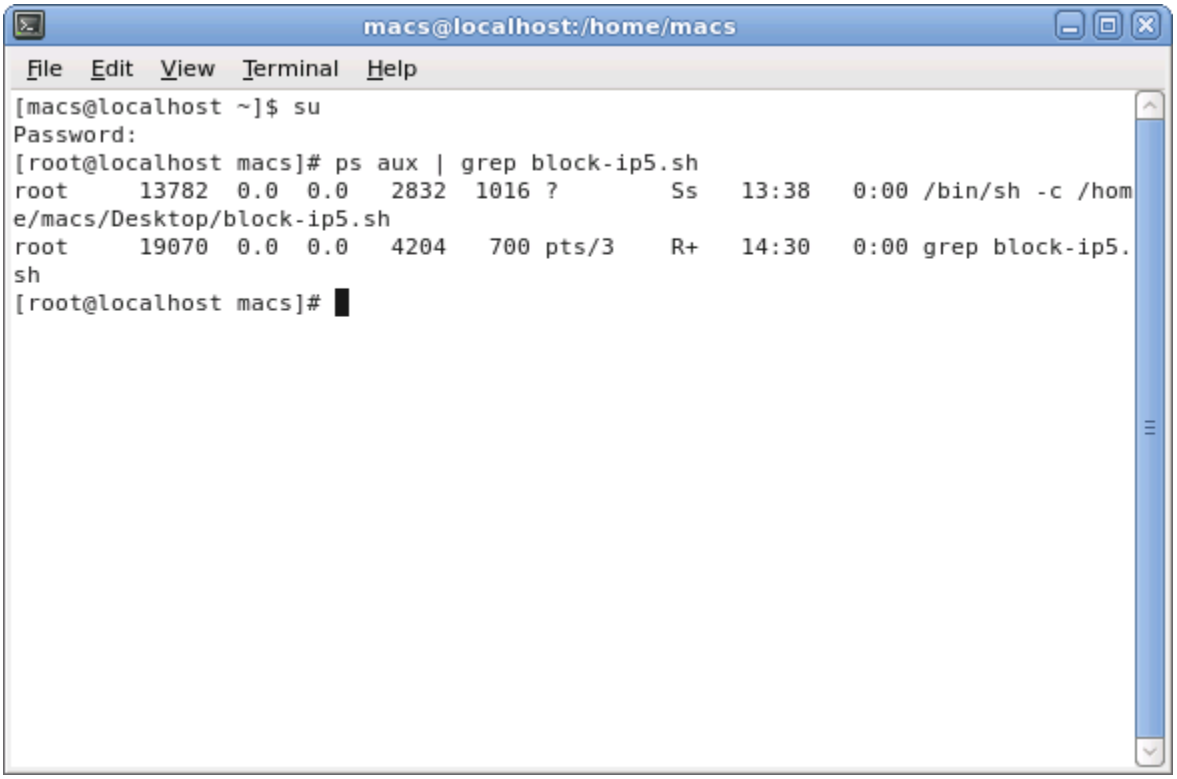

Figure 6 - block-ip5.sh listed in ps meaning the script is running

Server Clean Up

Following the tests, the server environment needs to be reverted to its previous configuration.

Open terminal and execute the following command:

```
ps aux | grep block-ip5.sh
```

Figure 7 - Show all processes even ones without controlling terminals and the user running each process that contain the string "block-ip5.sh"

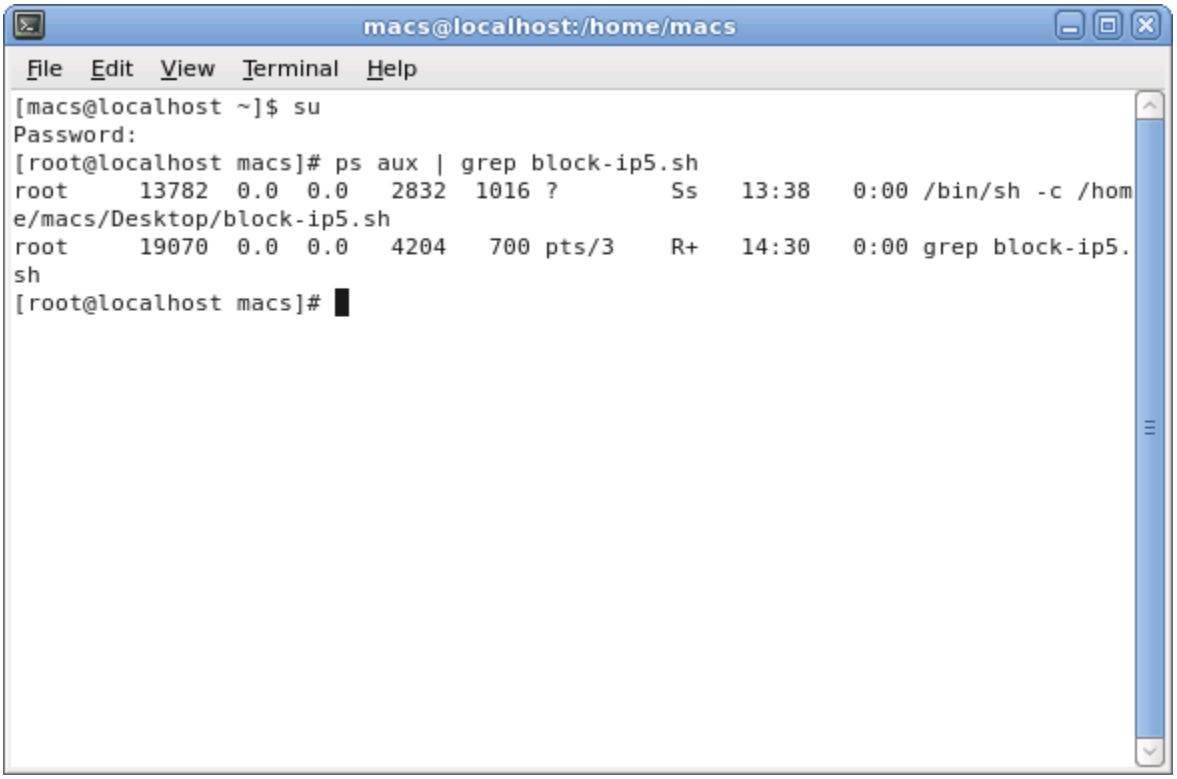

Figure 8 - block-ip5.sh listed in ps meaning the script is running

This will determine the process identifier for the process running block-ip5.sh

Execute the following command in terminal to kill the process:

```
su
<Your root password>
kill -9 13447
```

Figure 9 - Send SIGKILL (9) signal to process 13447 to end the process

Ensure that block-ip5.sh is no longer running by executing the following command in terminal:

```
ps aux | grep block-ip5.sh
```

Figure 10 - Show all processes even ones without controlling terminals and the user running each process that contain the string "block-ip5.sh"

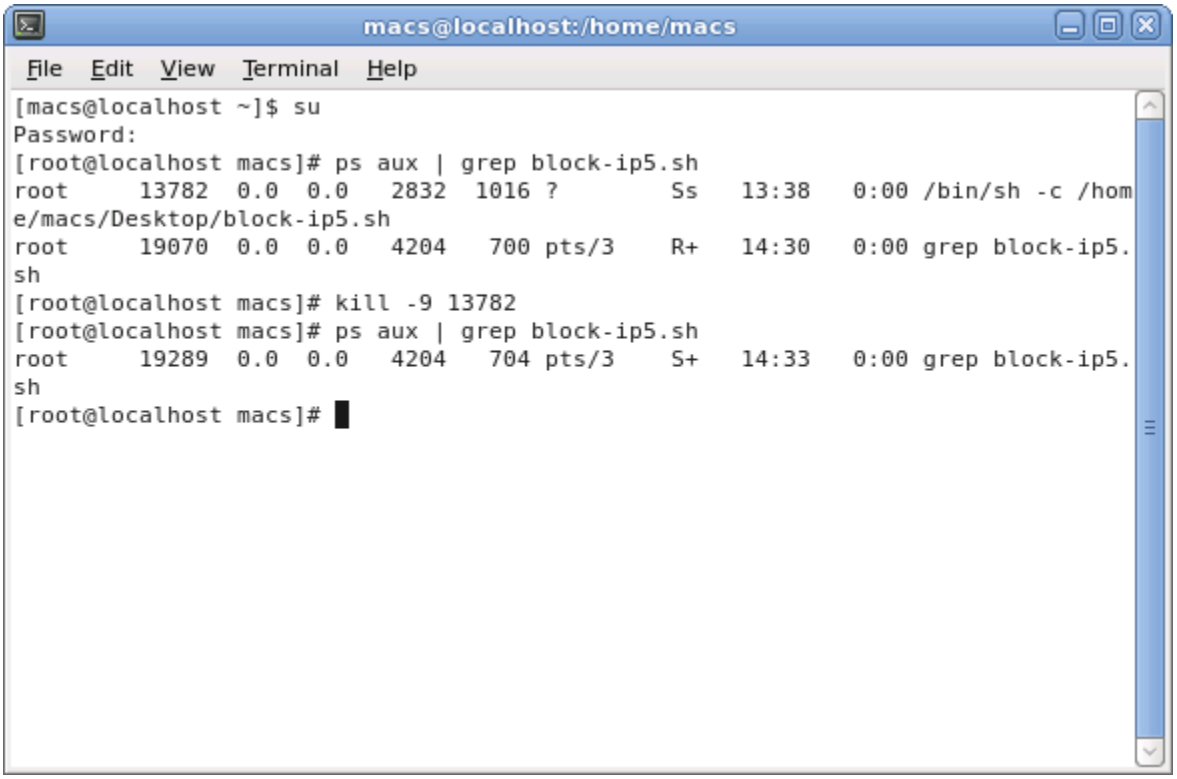

Figure 11 - block-ip5.sh is no longer listed in ps meaning the script has terminated

Test Cases

Rule #	Test Description	Tools Used	Expected Result	Pass/Fail
1	Four failed login attempts over a four day time span	sshd ssh crontab iptables	The failed login attempts will exceed the blocking threshold so the script adds an iptables rule to block the IP address, and a crontab rule to remove the iptables rule and the crontab rule itself after 30 minutes has passed	Pass. Detailed results are attached.
2	Three failed login attempts followed by one successful login followed by three failed login attempts	sshd ssh crontab iptables	The first three failed login attempts will be ignored by the script because of a successful login and the three failed login attempts after that will not exceed the blocking threshold so the script does not block the IP address	Pass. Detailed results are attached.

Test Case 1

The following test was run to see if failed login attempts spaced out over time are detected by the script and blocked by doing a failed login attempt a day for four days.

After the fourth unsuccessful login attempt, further login attempts timed out.

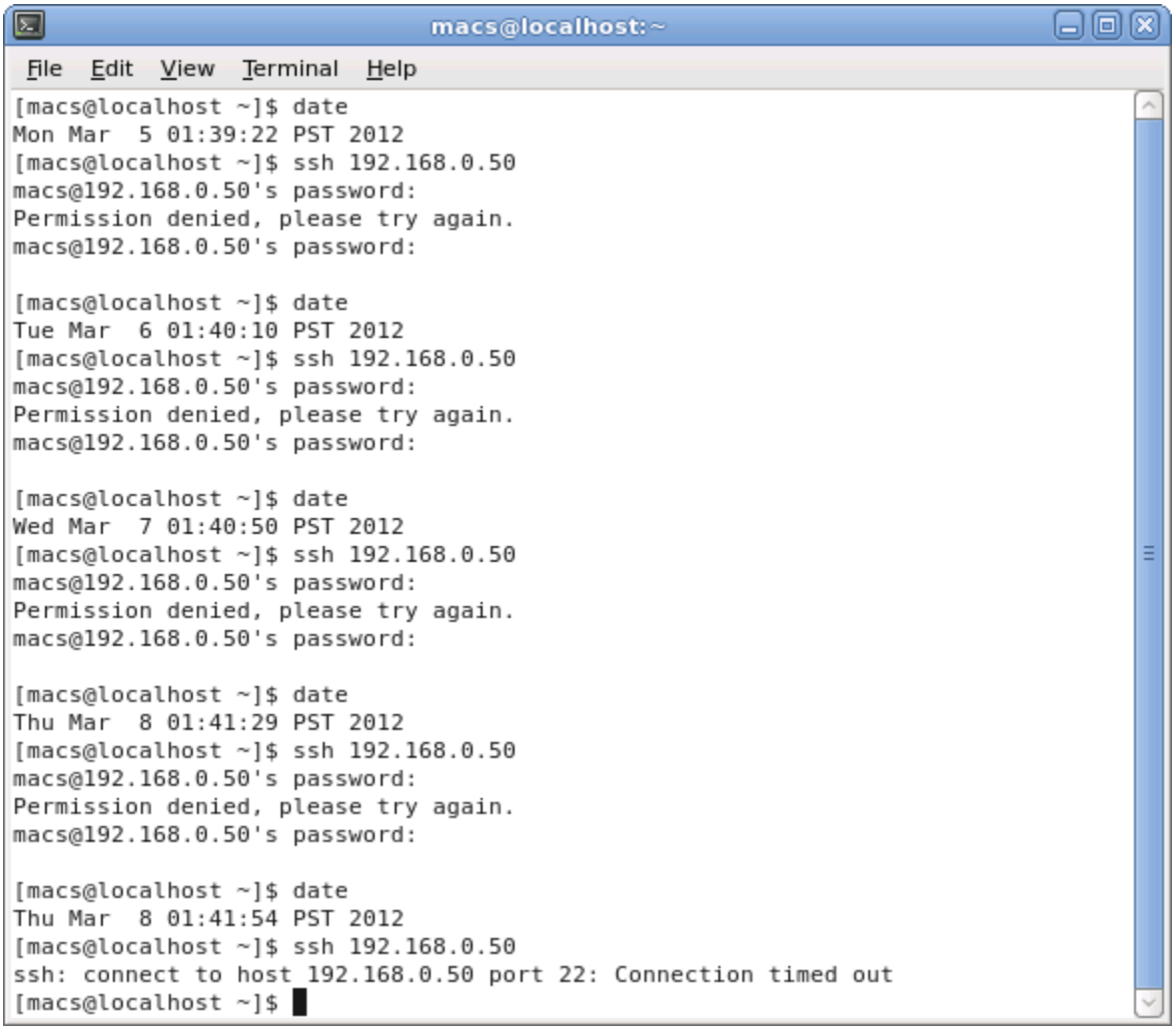

Figure 12 - Login attempts blocked after four unsuccessful tries

Further examination showed that the script added an iptables rule to drop all packets from the client IP address, a crontab entry to remove the iptables rule and the crontab entry itself after 30 minutes has passed, and an IP banned event in /var/log/secure.

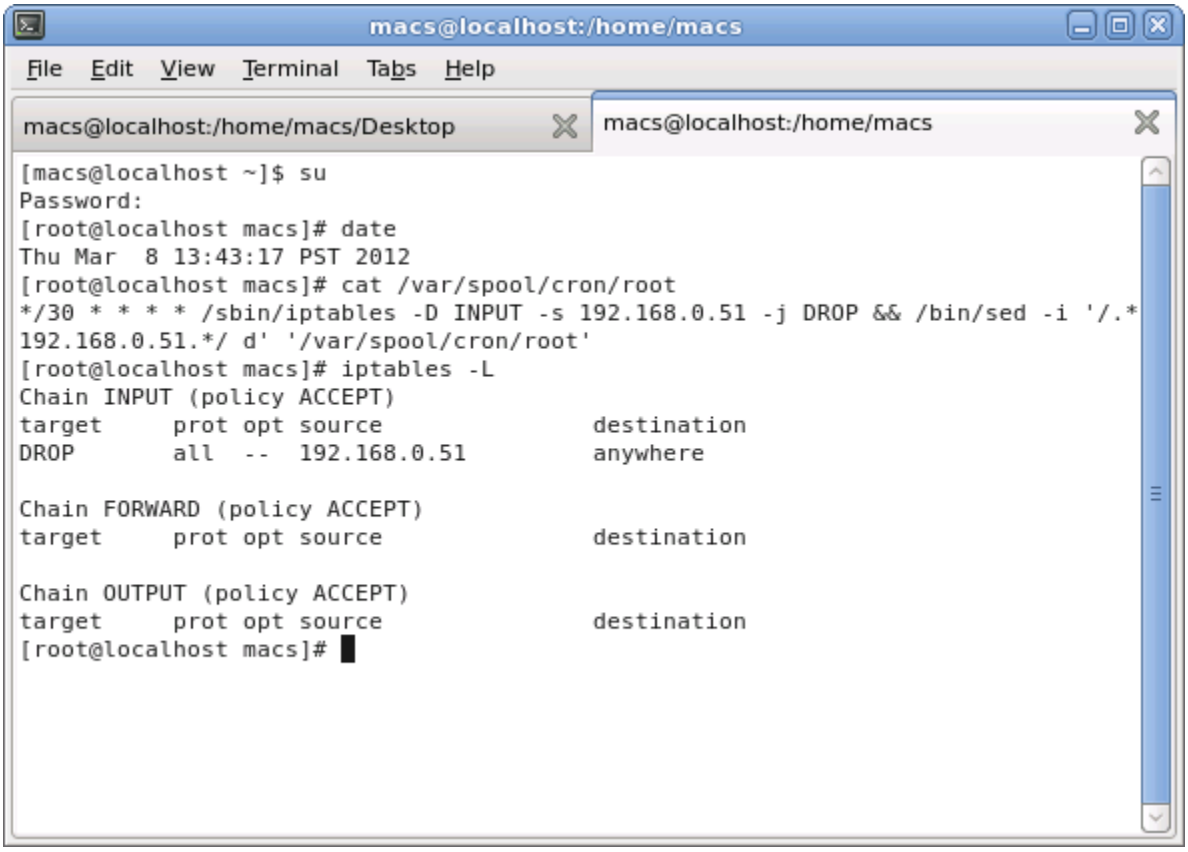

Figure 13 - iptables rule and crontab event created to block offending IP address and to remove the iptables rule and the crontab rule itself after 30 minutes

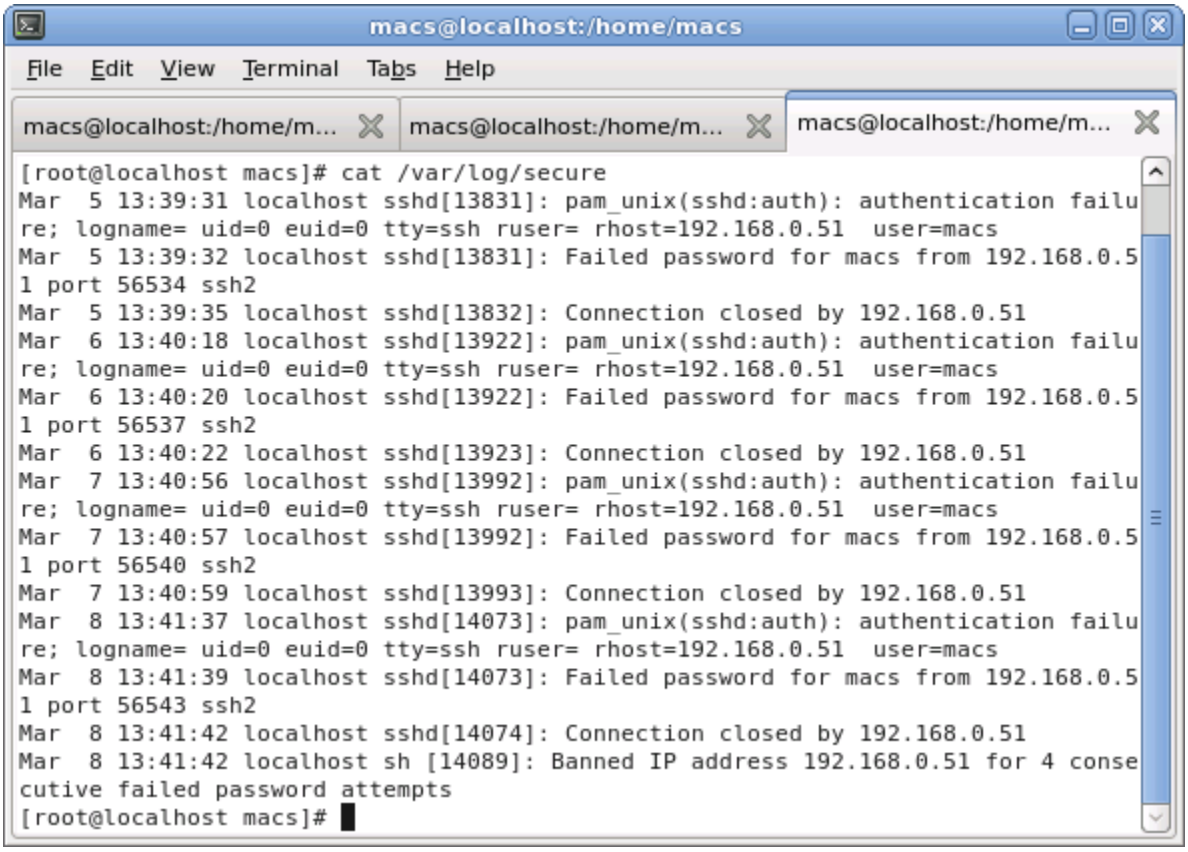

Figure 14 - Banned IP address event created in /var/log/secure

After 32 minutes has passed, the crontab rule removed the iptables rule and the crontab entry itself.

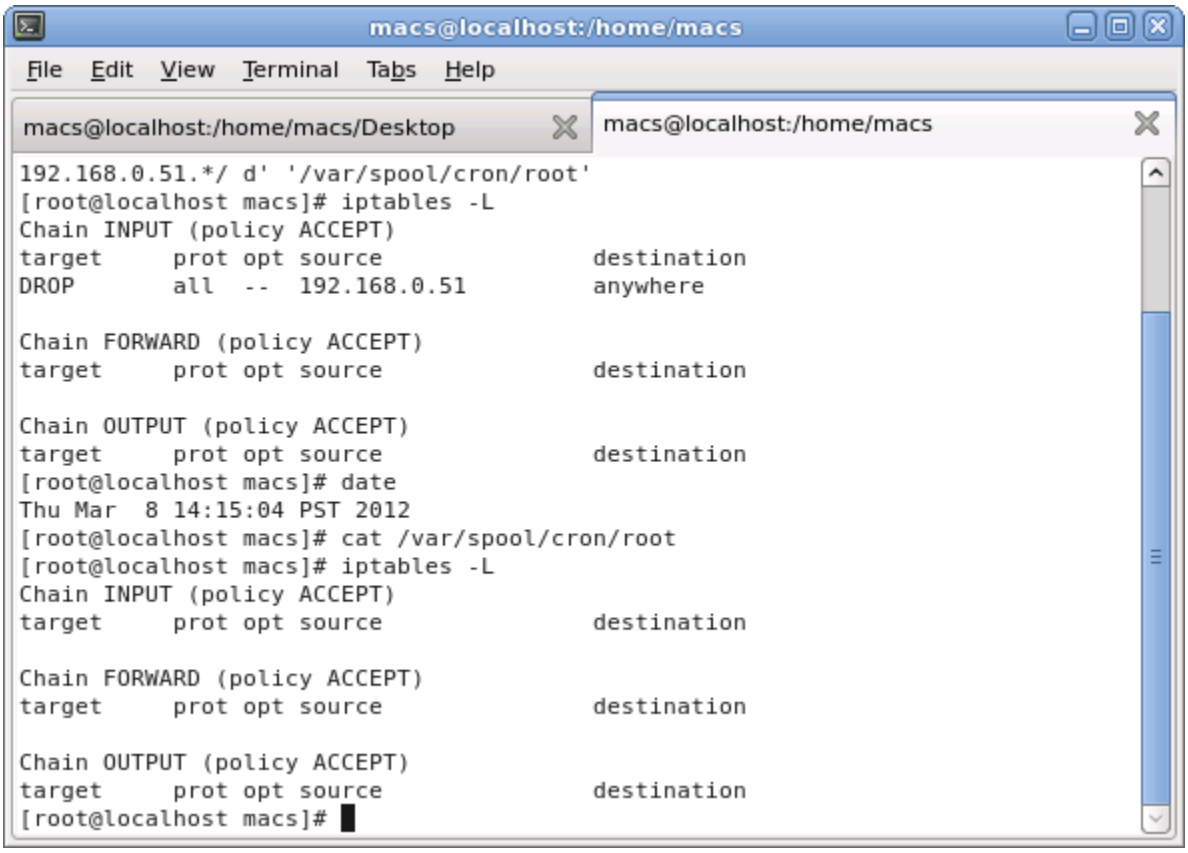

Figure 15 - iptables rule and crontab event removed after the blocking duration elapsed

Since the script successfully detected and blocked an IP address after exceeding the failed password threshold and unblocked the IP address after the block time duration has elapsed, the test passed.

Test Case 2

The following test was run to see if successful login attempts wipe the history of any previous failed login attempts for that IP address by doing three failed login attempts followed by one successful login followed by three failed login attempts.

After one successful login, the histories of the previous three failed login attempts are forgotten so the user can attempt to login another three times.

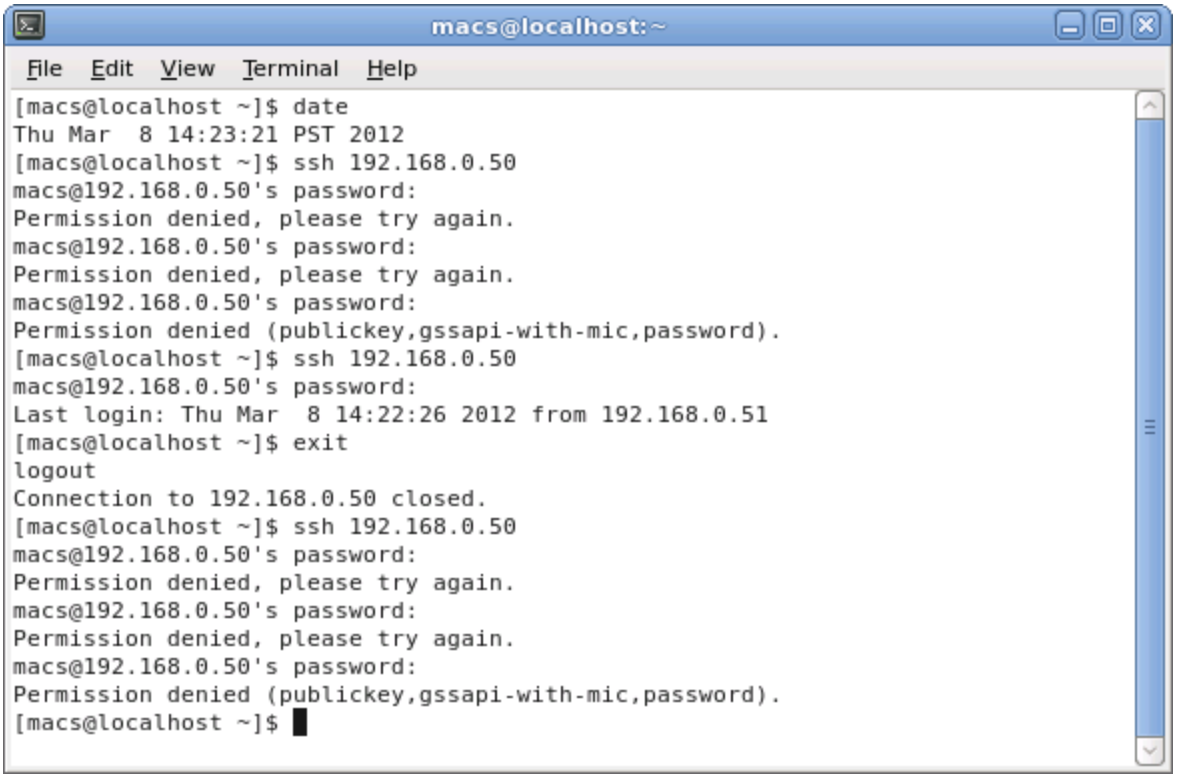

Figure 16 - Successful login removes history of the previous three failed login attempts

Further examination showed that the script did not add any iptables rules nor crontab entries and there are no IP banned events in /var/log/secure.

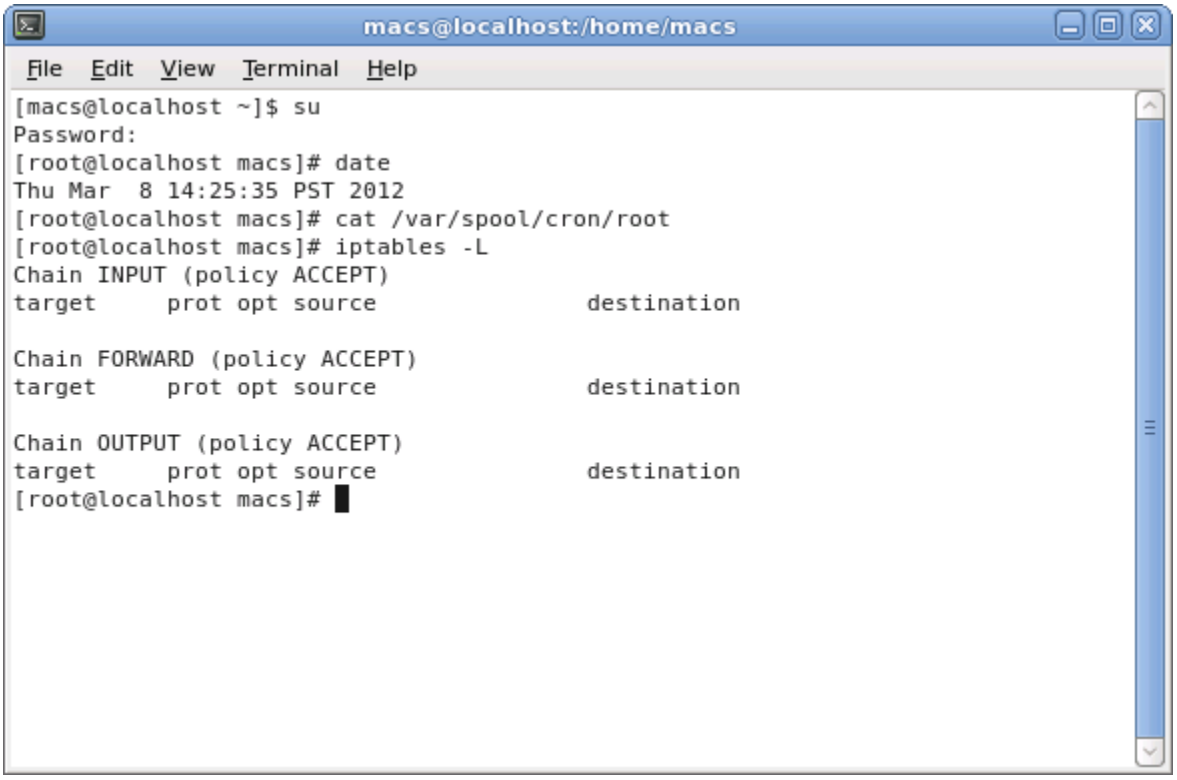

Figure 17 - No iptables blocking rules nor crontab events created by the script

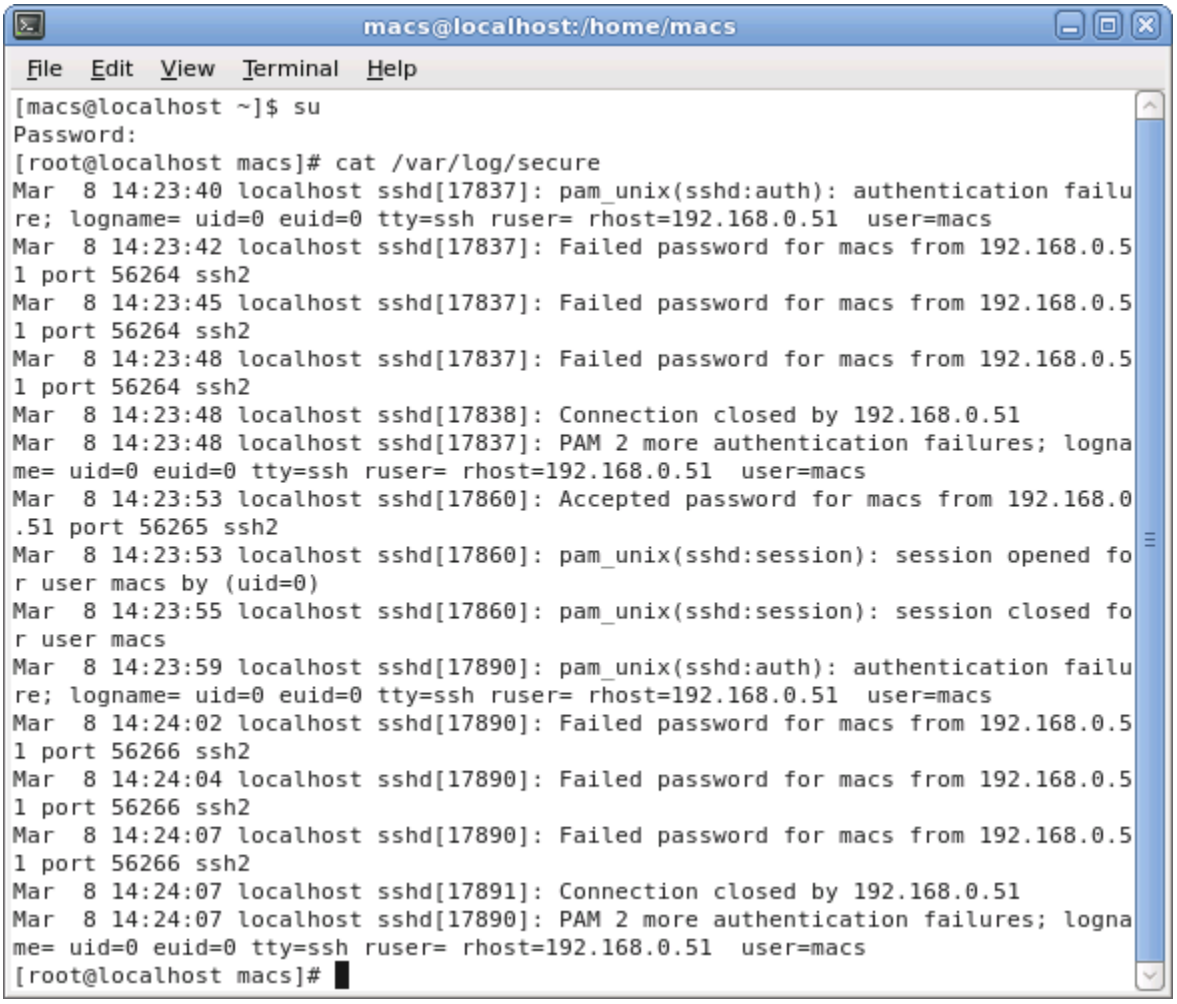

Figure 18 - No banned IP address event created in /var/log/secure

Since the script ignored the first three failed login attempts after a successful login, the test passed.