LxIPS

A Linux Based Intrusion Prevention System

By Ramzi Chennafi

Table of Contents

About the Program

This program was designed as a simple, customizable intrusion prevention system for Linux. Its core function it to watch authentication logs and respond to user specified network events by banning internet addresses from access.

Modification of program rules is done within the “rules.cfg”. Changing the rule file can be done by editing the configuration settings within LxIPS.rb. The changing of log files can also be done here.

The program allows each rule to specify a response done in IPTABLES, an attempt limit, a time to check between each attempts for attempt resetting and a timebanning value.

Several log files may also be watched, and if you refer to the readme, instructions for installing the program so that it does its work in the background after startup is included.

Included Files

Included in this package are the following files.

LxIPS.rb

src/manager.rb

src/user.rb

src/rule.rb

README

rules.cfg

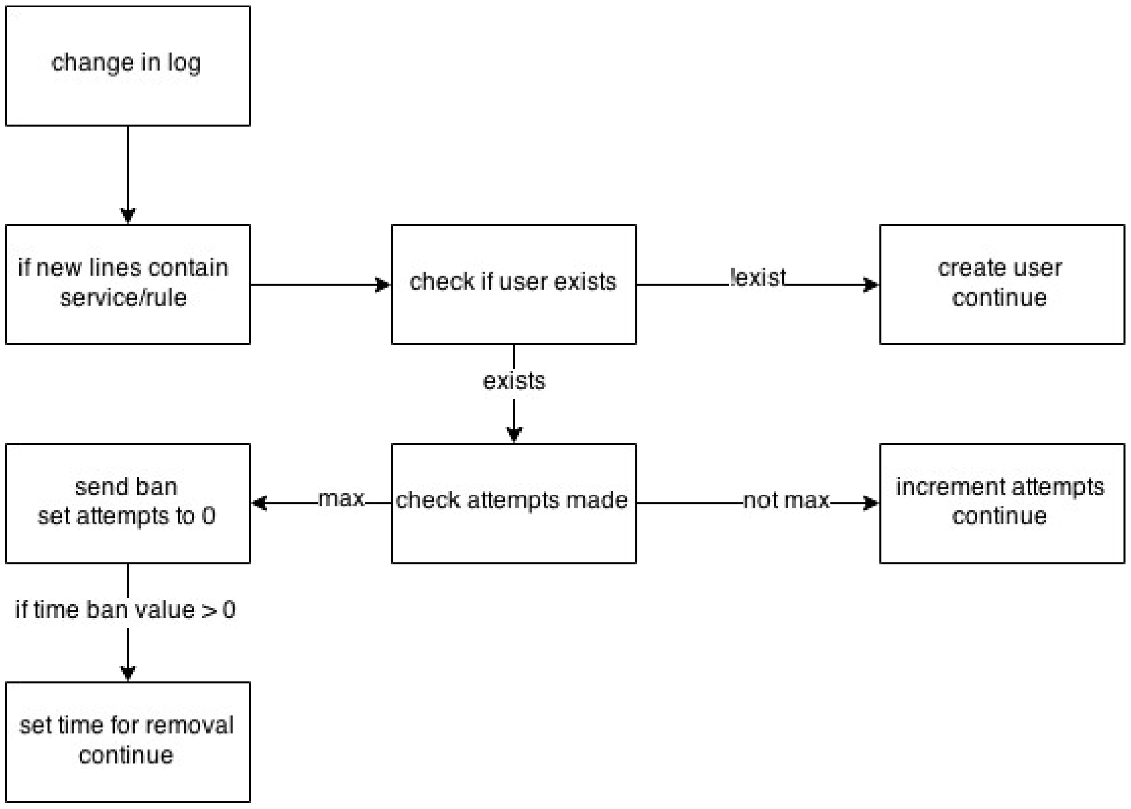
And this document

Execution

Please refer to the README for directions on usage and installation.

Design

System Overview


change in log
if new lines contain service/rule
check if user exists
!exist
create user continue
exists
check attempts made
max
send ban set attempts to 0
not max
increment attempts continue
if time ban value > 0
set time for removal continue

This diagram is a simple view of how the rule checking occurs in the program. The times and attempts are checked against the rules.cfg file.

System Diagram

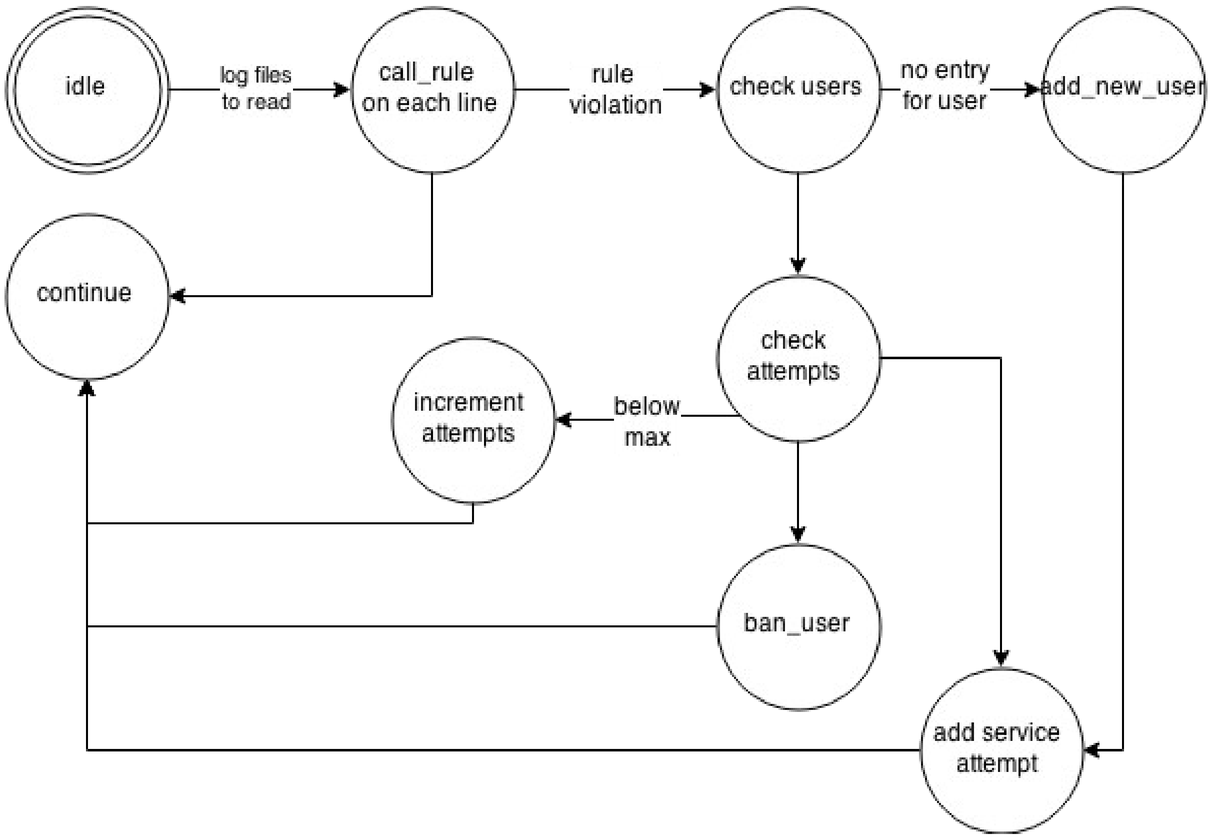
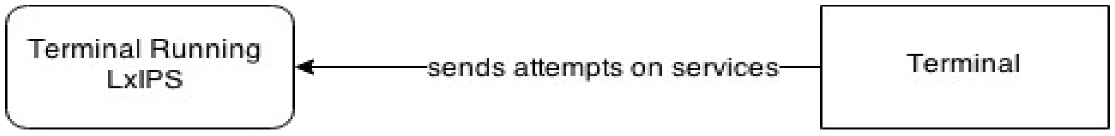


Table1

This is a program flow diagram. Programmatically, the application monitors a log file for changes and responds with a call rule on every new line. This program is single threaded, however at ban_user a sleeping thread is created to unban for time bans. It unbans once its sleep is over and closes.

Testing Setup



Above is a simple diagram depicting how our test setup ran. The two terminals share a subnet.

PseudoCode

Classes : User, Rule, Manager

Manager

Initialize()

```
setup_iptables()
open file and read each line
    if line is a rule
        create a rule object with it
```

setup_iptables()

```
Call iptables commands for LxIPS
```

add_new_user(ip, rule)

```
Create new user with ip
add_service_attempt on the rule passed
```

ban_user(ip, rule)

```
Ban user in iptables
set attempts for service the rule effects
set user status to banned
set the time they were banned
set the time they are to be unbanned
if user[ip].time_ban > 0
    create new thread using unban_user(ip, rule)
    print output
```

unban_user(ip, rule)

```
sleep(length of rule.time_ban)
unban user in iptables
```

call_rule(line)

```
For each line check for an entry for the service and event
    grab IP from line, store in ip
    if user at index ip does not exist
        add_new_user(ip, rule)
    else if user is valid
        switch(user[ip].attempts[service])
            case nil
                user[ip].add_service_attempt(rule.service)
            when (rule.attempts – 1)
                if time between attempts is valid
                    ban_user(ip, rule)
            when (1 to rule.attempts – 2)
                if time between attempts is valid
                    increment attempts on service for user
```

check_rules(file)

```
Call rule on each line of file
```

Rule

initialize(rule)

```
Split the rule for each :
store each parsed variable in its respective class variable
```

print_attempt()

```
Print the rule out
```

User

initialize(ip)

```
Create user with set ip
set default values
```

add_service_attempt(service)

```
Increment service.attempt
set_time_attempt(service)
print_attempt(service)
```

set_time_attempt(service)

```
Set the time of the last attempt on the service
```

check_time(attempt_time, service)

```
If attempt_time is passed the service attempt time limit
    return false
else return true
```

print_attempt(service)

```
Print the attempt on the service
```

self.get_time()

```
Get the hours
get the minutes
sum hours + minutes as minutes and return the number
```

Main Body

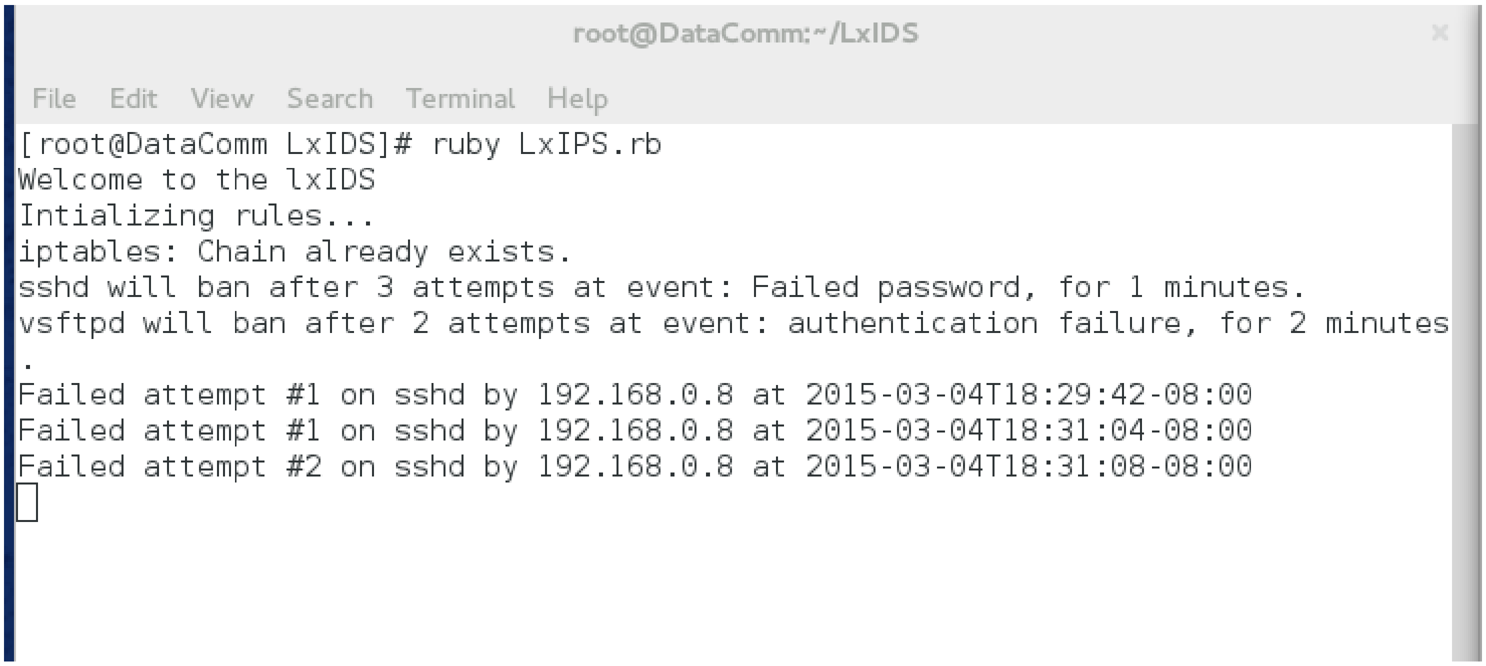
```
Create a new rule manager
for each log file
    open log file and seek to end
    add a notifier for modify on the log file with the callback call_rules
watch for notifiers
```

Testing

Test Number	Test Name	Test Description	Tools Used	Pass/Fail?
1	Timed Attempts	Check if an attempt done within the time limit is the only one counted.	*	Pass
2	Different Attempts	Check if an attempt on one service does not effect another service.	*	Pass
3	User Banning	Check if a user is banned when they max attempts on a service.	*	Pass
4	Time Bans	Check if a user banned on a time ban is unbanned.	*	Pass
5	Logging	Check if the logging for the program works.	*	Pass
6	Rule Configurability	Check if user can create and change rules.	*	Pass
7	Starts on Boot	Check if the program starts on boot.	*, crontab, ps	Pass

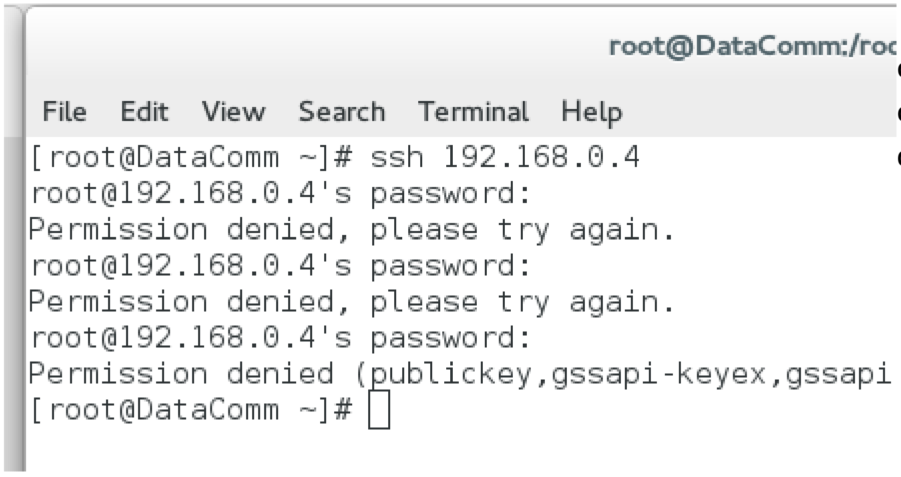
* = the program itself

Test 1 : Timed Attempts



```
[root@DataComm LxIDS]# ruby LxIPS.rb
Welcome to the lxIDS
Intializing rules...
iptables: Chain already exists.
sshd will ban after 3 attempts at event: Failed password, for 1 minutes.
vsftpd will ban after 2 attempts at event: authentication failure, for 2 minutes
.
Failed attempt #1 on sshd by 192.168.0.8 at 2015-03-04T18:29:42-08:00
Failed attempt #1 on sshd by 192.168.0.8 at 2015-03-04T18:31:04-08:00
Failed attempt #2 on sshd by 192.168.0.8 at 2015-03-04T18:31:08-08:00
```

Above you can see that a terminal at 192.168.0.8 made several attempts at accessing SSH. The rule for ssh here has a time limit of 1 minute between attempts. The second attempt did not increase the attempts because it occurred 1 minute and 22 seconds after the first, 22 seconds passed the attempt time limit. This shows that timed attempts is working properly.

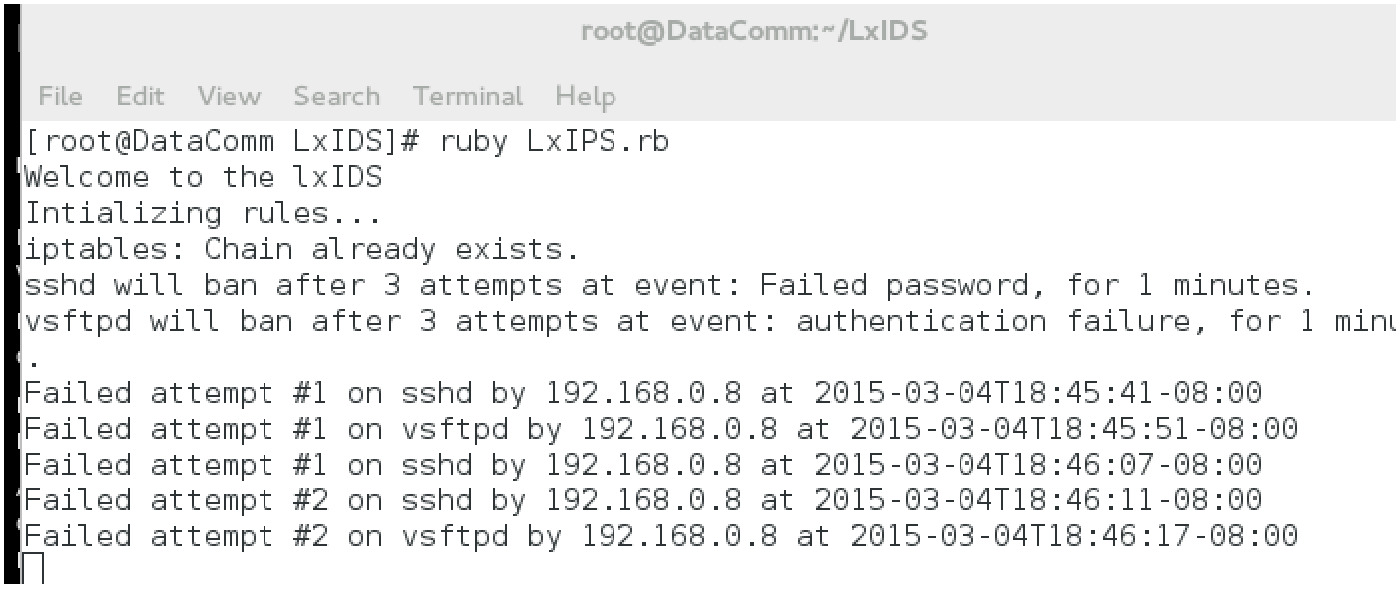


```
[root@DataComm ~]# ssh 192.168.0.4
root@192.168.0.4's password:
Permission denied, please try again.
root@192.168.0.4's password:
Permission denied, please try again.
root@192.168.0.4's password:
Permission denied (publickey,gssapi-keyex,gssapi
[root@DataComm ~]#
```

The terminal here is the computer attempting to ssh, as you can see, their permission is being denied with each failed attempt.

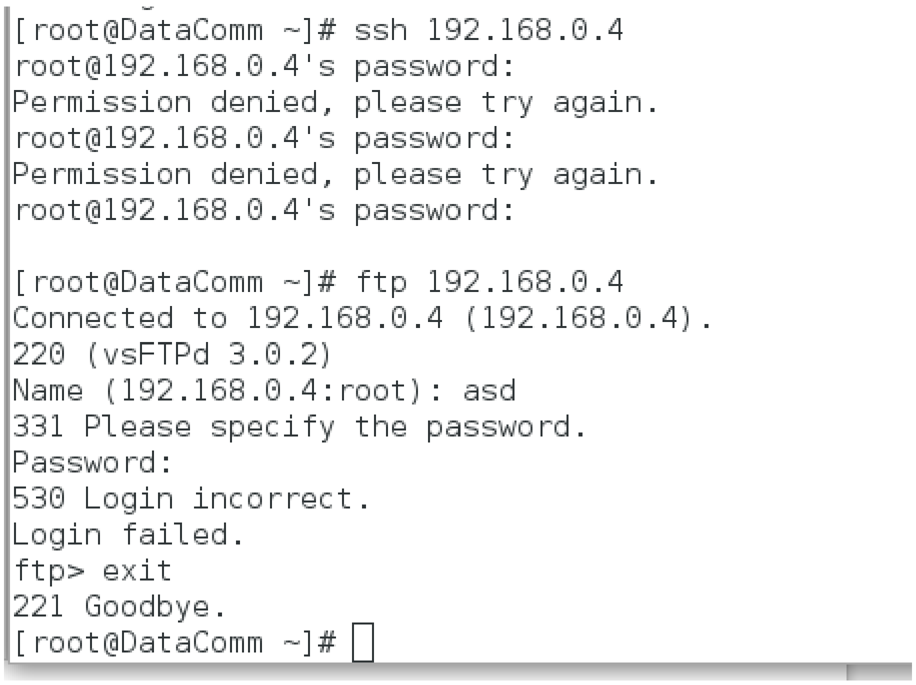
Test 2 : Different Attempts

Here I will be testing what happens when a user makes invalid attempts on several services.



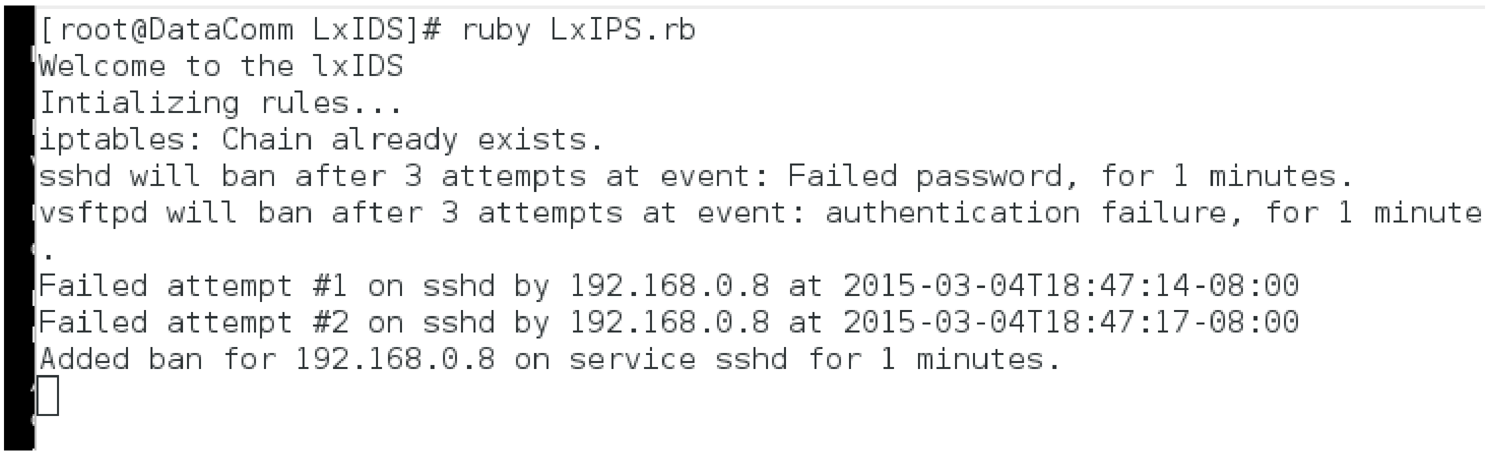
```
[root@DataComm LxIDS]# ruby LxIPS.rb
Welcome to the lxIDS
Intializing rules...
iptables: Chain already exists.
sshd will ban after 3 attempts at event: Failed password, for 1 minutes.
vsftpd will ban after 3 attempts at event: authentication failure, for 1 minu
.
Failed attempt #1 on sshd by 192.168.0.8 at 2015-03-04T18:45:41-08:00
Failed attempt #1 on vsftpd by 192.168.0.8 at 2015-03-04T18:45:51-08:00
Failed attempt #1 on sshd by 192.168.0.8 at 2015-03-04T18:46:07-08:00
Failed attempt #2 on sshd by 192.168.0.8 at 2015-03-04T18:46:11-08:00
Failed attempt #2 on vsftpd by 192.168.0.8 at 2015-03-04T18:46:17-08:00
```

As you can see here, several attempts were made on both ftp and ssh. The attempts on each had no effect on the others. This shows its working correctly. If you look below you'll see the terminal on the outside computer attempting these events.



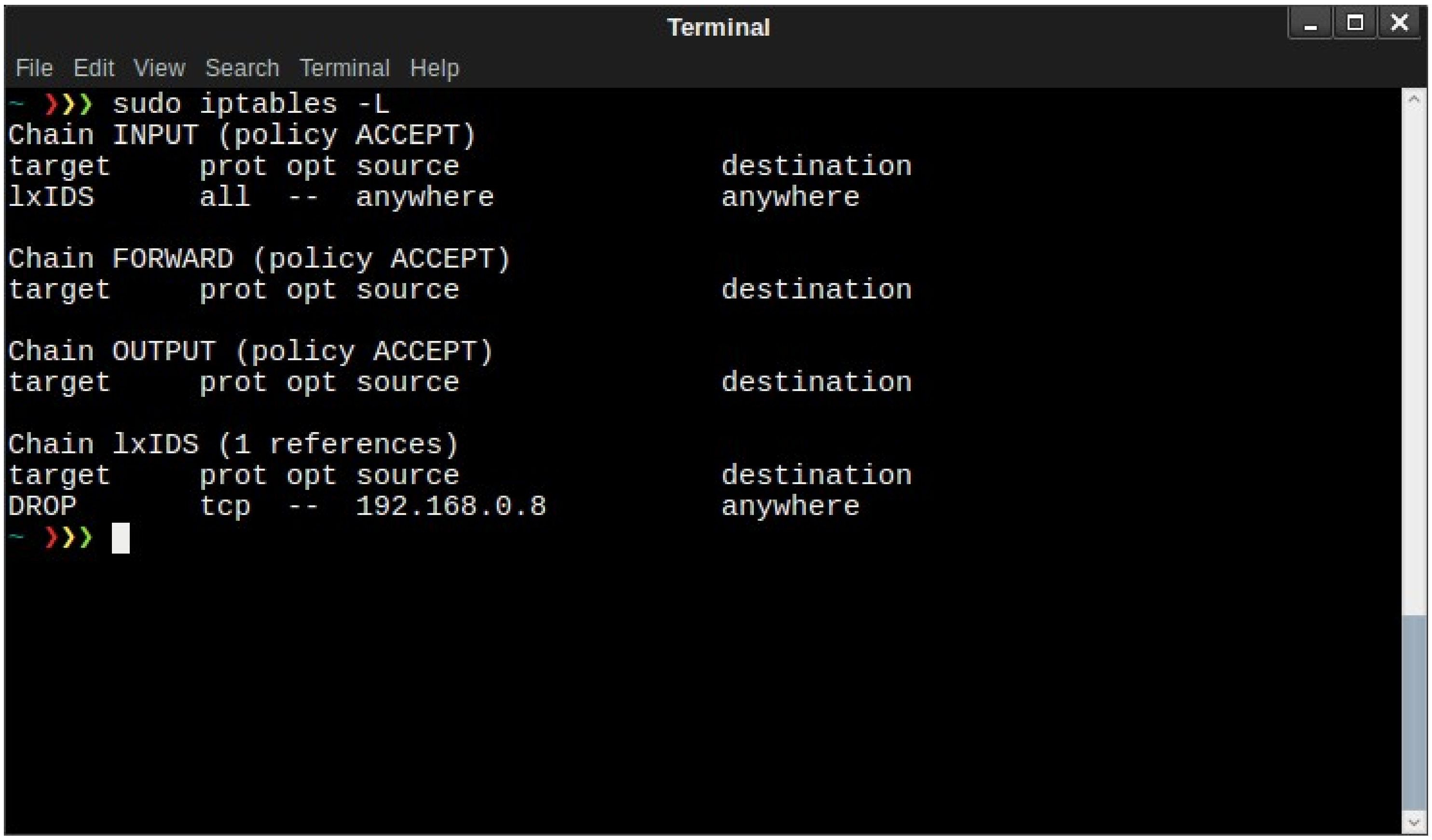
```
[root@DataComm ~]# ssh 192.168.0.4
root@192.168.0.4's password:
Permission denied, please try again.
root@192.168.0.4's password:
Permission denied, please try again.
root@192.168.0.4's password:

[root@DataComm ~]# ftp 192.168.0.4
Connected to 192.168.0.4 (192.168.0.4).
220 (vsFTPd 3.0.2)
Name (192.168.0.4:root): asd
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp> exit
221 Goodbye.
[root@DataComm ~]#
```

Test 3 : User Banning



```
[root@DataComm LxIDS]# ruby LxIPS.rb
Welcome to the lxIDS
Intializing rules...
iptables: Chain already exists.
sshd will ban after 3 attempts at event: Failed password, for 1 minutes.
vsftpd will ban after 3 attempts at event: authentication failure, for 1 minute
.
Failed attempt #1 on sshd by 192.168.0.8 at 2015-03-04T18:47:14-08:00
Failed attempt #2 on sshd by 192.168.0.8 at 2015-03-04T18:47:17-08:00
Added ban for 192.168.0.8 on service sshd for 1 minutes.
```

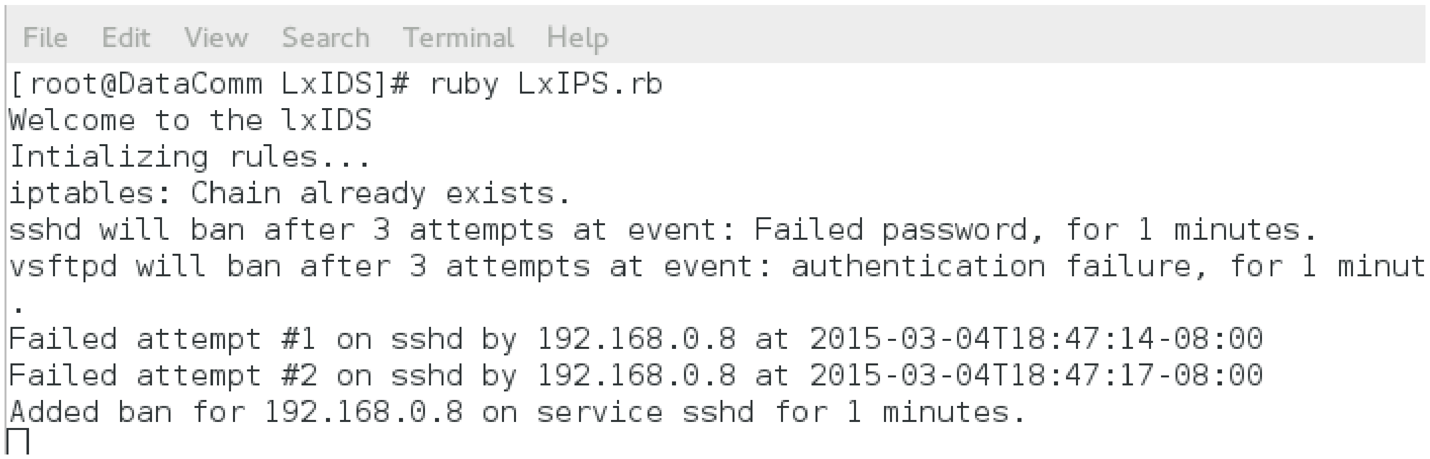
In the above picture you can see the program banning 192.168.0.8 for 3 failed attempts on the system. If we refer to IPTABLES we should see this entry.



```
~ ❯❯❯ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
lxIDS      all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain lxIDS (1 references)
target     prot opt source               destination
DROP       tcp  --  192.168.0.8          anywhere
~ ❯❯❯
```

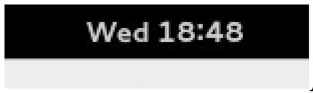
As you can see, the entry is in IPTABLES, this shows that the banning was successful.

Test 4 : Time Bans

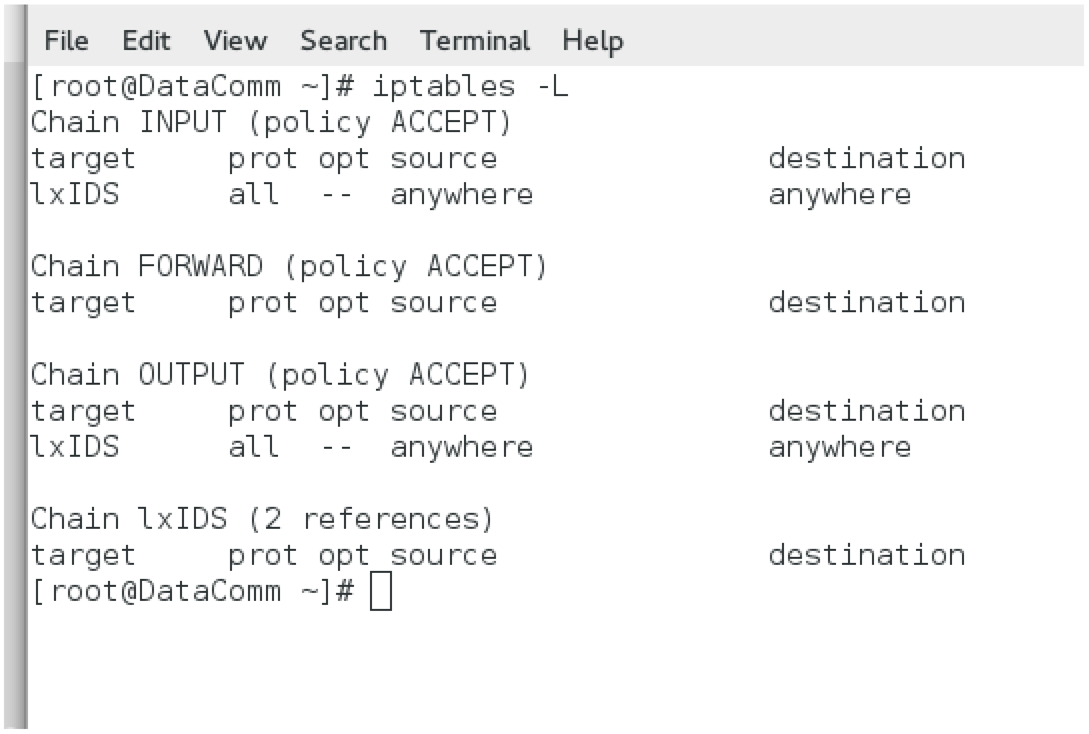


```
[root@DataComm LxIDS]# ruby LxIPS.rb
Welcome to the lxIDS
Intializing rules...
iptables: Chain already exists.
sshd will ban after 3 attempts at event: Failed password, for 1 minutes.
vsftpd will ban after 3 attempts at event: authentication failure, for 1 minut
.
Failed attempt #1 on sshd by 192.168.0.8 at 2015-03-04T18:47:14-08:00
Failed attempt #2 on sshd by 192.168.0.8 at 2015-03-04T18:47:17-08:00
Added ban for 192.168.0.8 on service sshd for 1 minutes.
```

In this test, the user made several failed attempts on ssh, take note that the ban occurred just after 18:47. If we look to the system time at this moment we see that its a minute past the banning time.



And if we look to IPTABLES, we see the entry for the ban is no longer there.



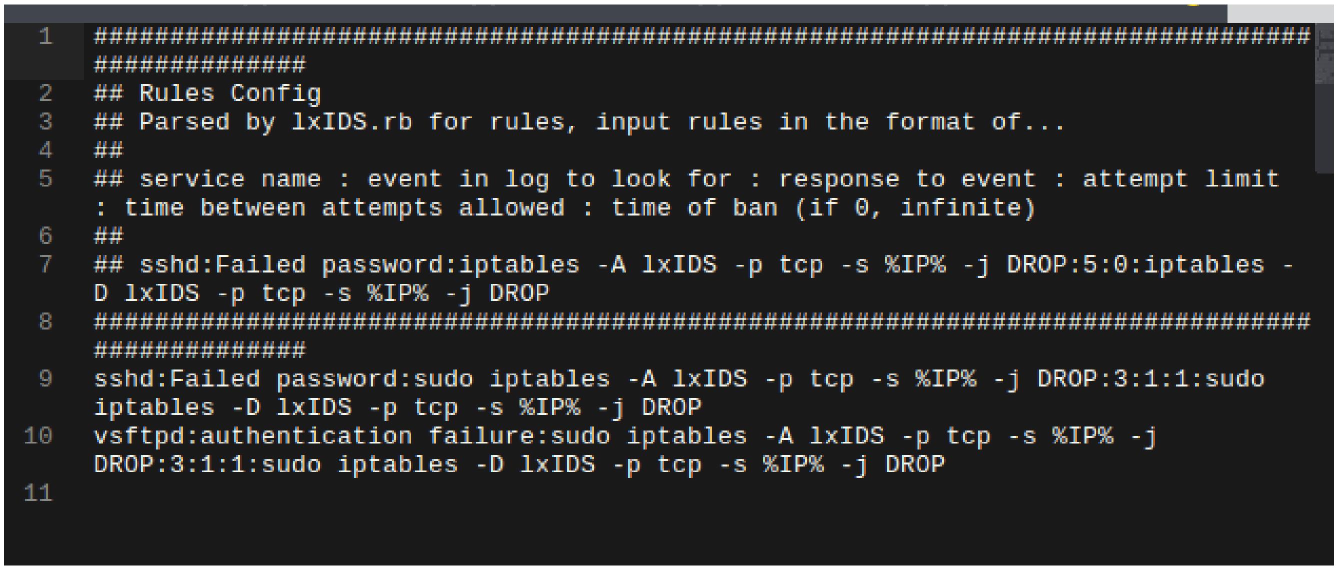
```
[root@DataComm ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
lxIDS      all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
lxIDS      all  --  anywhere             anywhere

Chain lxIDS (2 references)
target     prot opt source               destination
[root@DataComm ~]# 
```

The test is successful.

Test 5 : Logging

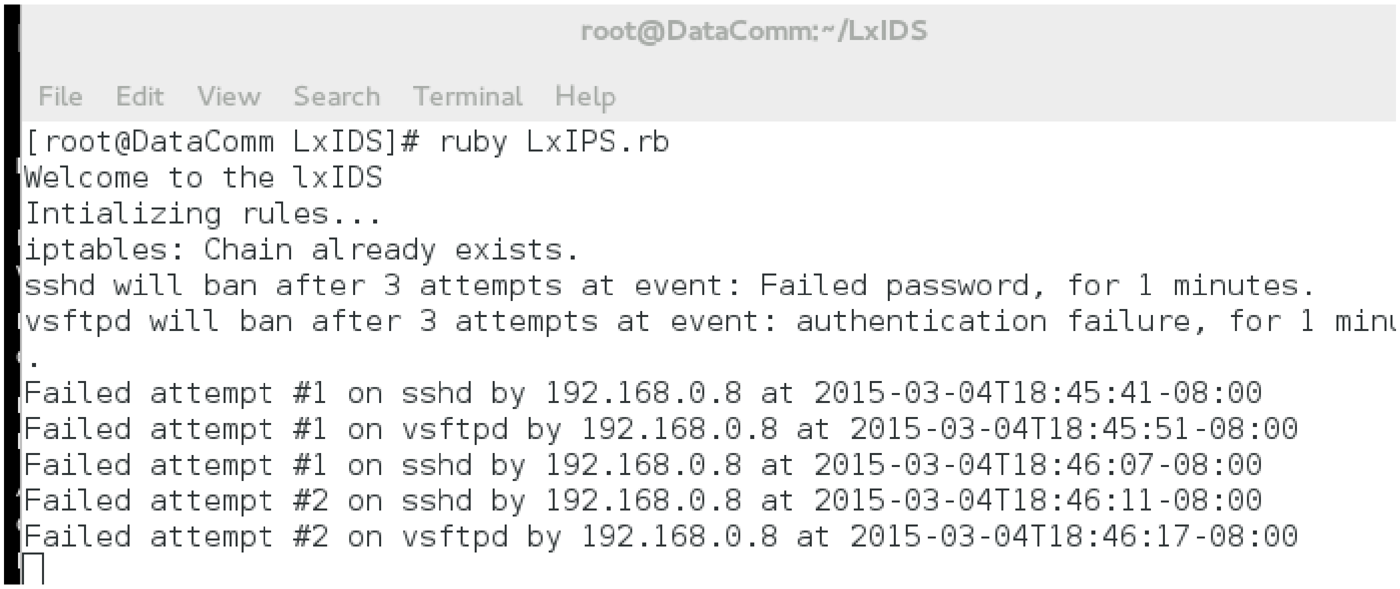
For this test, I would ask you would refer to previous tests. The output printed is valid in each terminal, the only requirement to get logging is to add an >> to a log file when referencing the command, since we know output redirection works – logging must work.

Test 6 : Rule Configurability



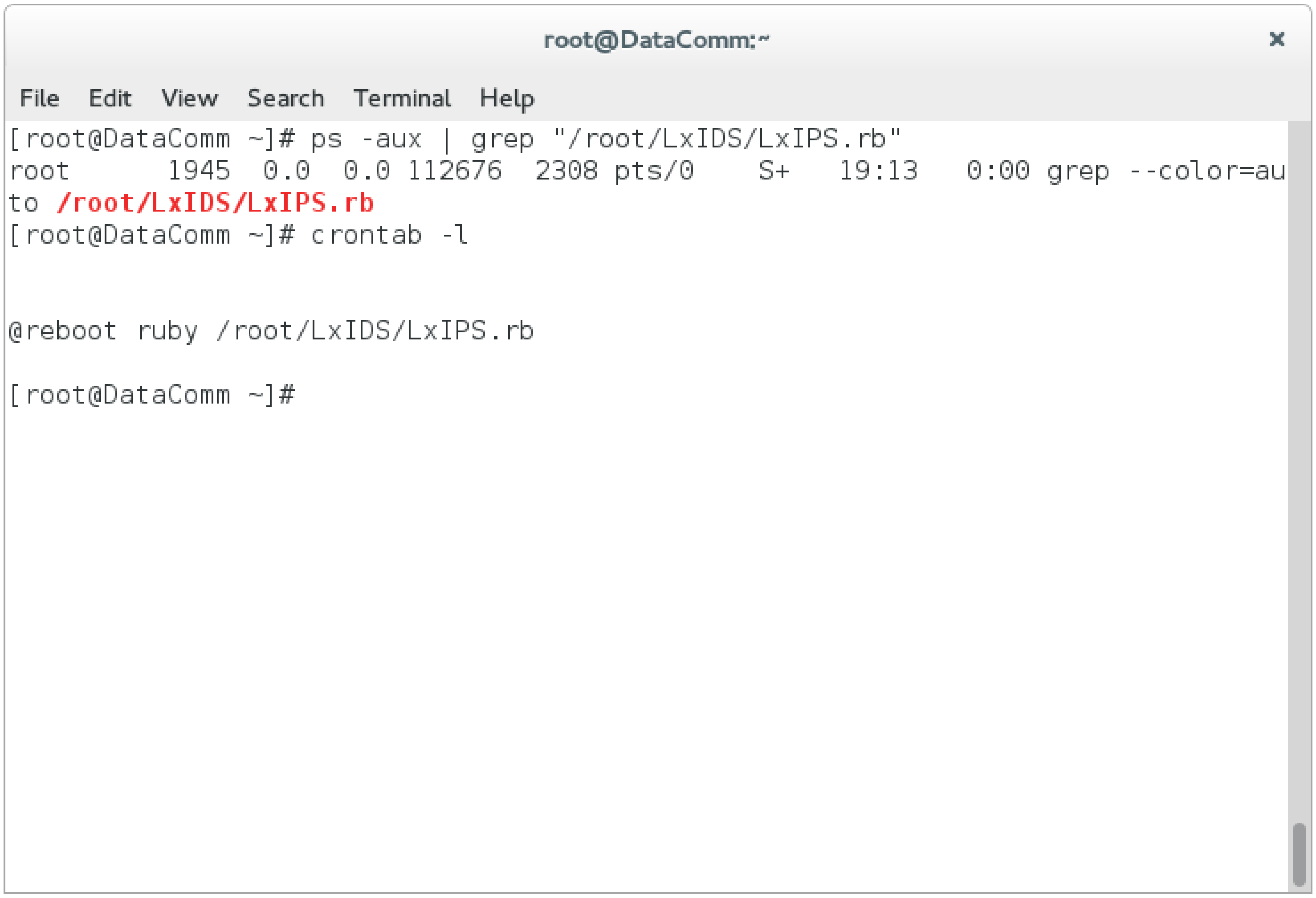
```
##############################################################################################
## Rules Config
## Parsed by lxIDS.rb for rules, input rules in the format of...
##
## service name : event in log to look for : response to event : attempt limit : time between attempts allowed : time of ban (if 0, infinite)
##
## sshd:Failed password:iptables -A lxIDS -p tcp -s %IP% -j DROP:5:0:iptables -D lxIDS -p tcp -s %IP% -j DROP
##############################################################################################
sshd:Failed password:sudo iptables -A lxIDS -p tcp -s %IP% -j DROP:3:1:1:sudo iptables -D lxIDS -p tcp -s %IP% -j DROP
vsftpd:authentication failure:sudo iptables -A lxIDS -p tcp -s %IP% -j DROP:3:1:1:sudo iptables -D lxIDS -p tcp -s %IP% -j DROP

```

As you can see here we have two rules inputted into our rule config file. If we look to the program startup we should see these rules printed to the terminal. Also take note that in the bottom picture, attempts were made on both services. The test is a success.



```
[root@DataComm LxIDS]# ruby LxIPS.rb
Welcome to the lxIDS
Intializing rules...
iptables: Chain already exists.
sshd will ban after 3 attempts at event: Failed password, for 1 minutes.
vsftpd will ban after 3 attempts at event: authentication failure, for 1 minu
.
Failed attempt #1 on sshd by 192.168.0.8 at 2015-03-04T18:45:41-08:00
Failed attempt #1 on vsftpd by 192.168.0.8 at 2015-03-04T18:45:51-08:00
Failed attempt #1 on sshd by 192.168.0.8 at 2015-03-04T18:46:07-08:00
Failed attempt #2 on sshd by 192.168.0.8 at 2015-03-04T18:46:11-08:00
Failed attempt #2 on vsftpd by 192.168.0.8 at 2015-03-04T18:46:17-08:00
```

Test 7 : Starts on Reboot



```
[root@DataComm ~]# ps -aux | grep "/root/LxIDS/LxIPS.rb"
root      1945  0.0  0.0 112676  2308 pts/0    S+   19:13   0:00 grep --color=au
to /root/LxIDS/LxIPS.rb
[root@DataComm ~]# crontab -l


@reboot ruby /root/LxIDS/LxIPS.rb

[root@DataComm ~]#
```

In this test, we will be checking if the setup for this program properly allows for it to start on reboot for monitoring. I have just rebooted the machine, as you can see from ps, LxIPS is running. You can also see the crontab entry for the start at boot. This was successful.