Chris Wood
A00741285

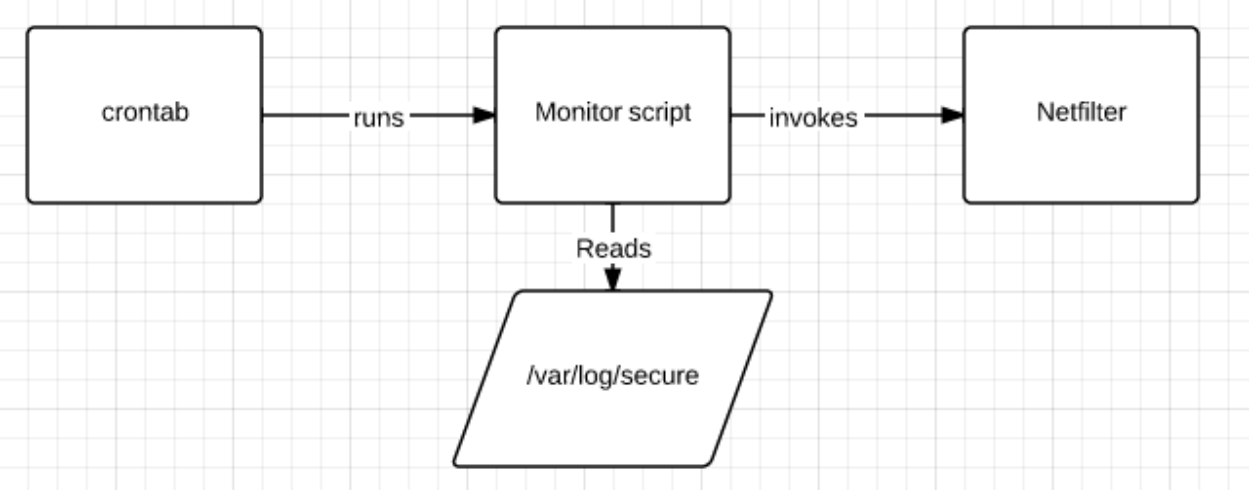
COMP8006 Assignment 3

Objective

To design, implement and test a simple monitor application that will detect password guessing attempts against a service and block that IP using Netfilter.

Implementation design

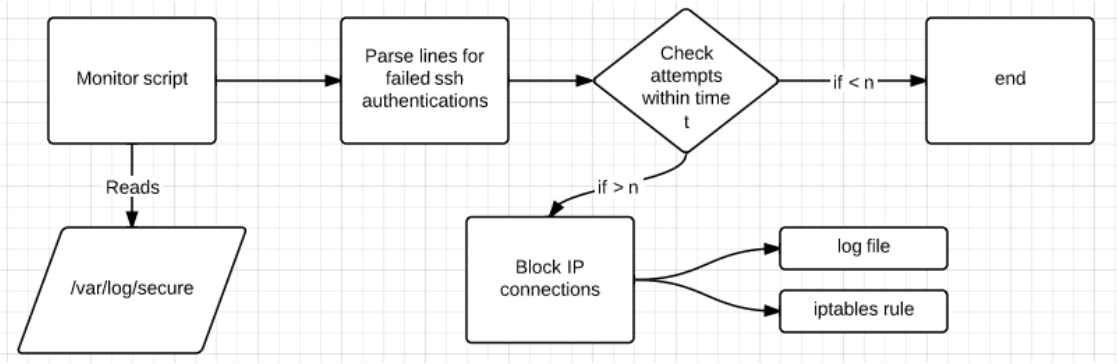
Monitor application will be ruby script file that will be invoked by crontab to look for ssh authentication failures in /var/log/secure



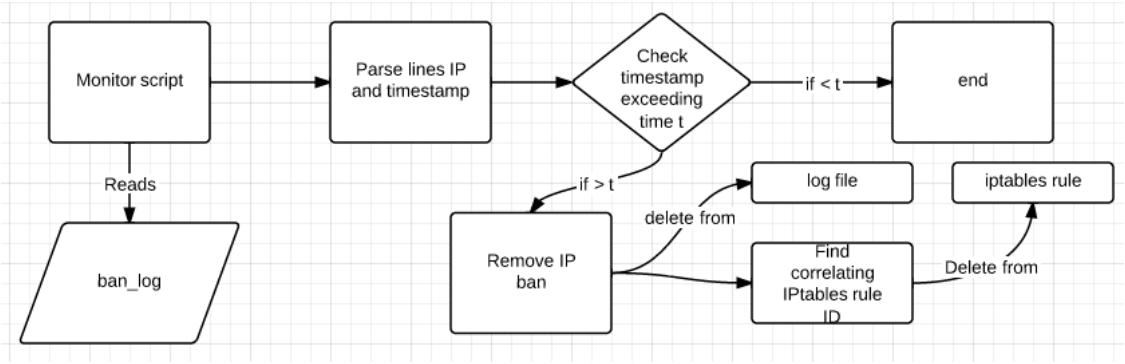
Monitor script will be composed of 2 parts

1. Check /var/log/secure
2. Check its own log

1. Monitor script with parse /var/log/secure line by line looking for failed ssh authentication attempts. If found, it will record the instance internally until done parsing. If there are a user specified number of attempts within a user specified time period, then the script will create an iptables rule to block all incoming connections from the IP the authentications were coming from. IP will also be added to the scripts own log with timestamp.



2. Monitor script checks own log to see if a banned IP has been blocked access for a user specified time, if the time frame is exceeded, then the IP will be removed from the internal log and the iptables rule deleted.



When looking at /var/log/secure, there was a pattern with the string “Failed password” coming from ssh authentications

```
[root@selenium comp8006-assign3]# cat /var/log/secure | grep "Failed password"
Mar  6 11:34:42 selenium sshd[39249]: Failed password for root from 192.168.15.132 po
rt 53045 ssh2
Mar  6 11:35:35 selenium sshd[39301]: Failed password for root from 192.168.15.132 po
rt 53046 ssh2
Mar  6 11:35:39 selenium sshd[39301]: Failed password for root from 192.168.15.132 po
rt 53046 ssh2
Mar  6 12:47:16 selenium sshd[39574]: Failed password for invalid user bob from 192.1
68.15.132 port 53055 ssh2
[root@selenium comp8006-assign3]#
```

Each line of the log file based from this filter had everything I needed to process my script

- Date
- Time
- Client IP address

Based on needing to use this information, I decided that the script will be written in Ruby for its string parsing capabilities. I was also interesting in giving Ruby a try.

Implementation functionality

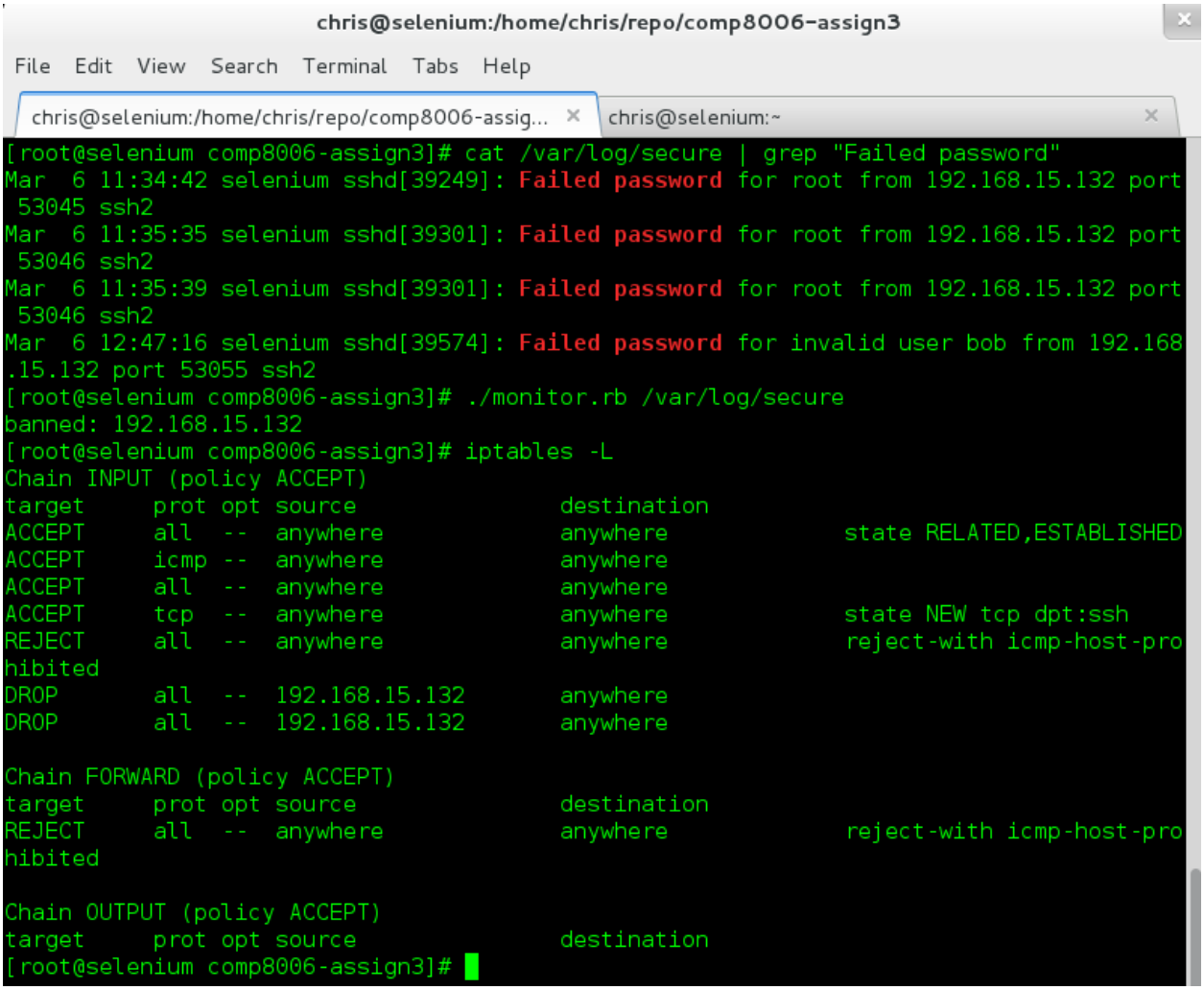
I will not be including all my code in this document as it is provide as source but I wanted to highlight the following

```ruby
#Object used for collecting valuable information from parse log lines
class Attempt
        def initialize(month, day, time, ip)
          @month        = month
          @day          = day
          @time         = time
          @ip           = ip
        end
        def month
          @month
        end
        def day
          @day
        end
        def time
          @time
        end
        def ip
          @ip
        end

        #Converts shorthand month strings to numerical numbers
        def monthToDig(month)
                case month
                when "Jan"
                        return 1
                when "Feb"
                        return 2
                when "Mar"
                        return 3
                when "Apr"
                        return 4
                end
                #etc.....
        end

        #Returns a time object constructed from an attempt object's local variables
        def to_time
                hms = @time.split(pattern=":")
                t = Time.new(Time.now.year, monthToDig(@month), @day.to_i, hms[0].to_i, hms[1].to_i, hms[2].to_i)
                return t.to_i
        end
end
```

Creating an object out of the parsed lines was helpful for me to keep track of each line of attempt and be able to access them in an easier manner. The method *to_time* returns the timestamp as the number of seconds from the Epoch[1]. I thought this would be a good method for comparing the time values easily.

[1] http://www.ruby-doc.org/core-2.0/Time.html#method-i-to_i

Testing



Invoking the script finds the 3 failed password attempts from 192.168.15.132 and adds a rule in to iptables to drop all traffic.

Final Notes

Submitted version of script does not have the functionality of time based removals of IPs as the implementation was not fully completed. The code for this is included but does not function as intended; it is comment out from being called.

The script is meant to be put in the crontab to constantly run and get for failed password attempts. The crontab entry would look as follows

```
*/5 * * * * /home/chris/repo/comp8006-assign3/monitor.rb
```