Computer Systems Technology

British Columbia Institute of Technology

COMP 8006 - Assignment3- Testing

Albert Huang &

Aiyan Ma

Feb 20, 2018

Table of Contents

1. Monitor Testing

1.1 Test Outline

Rule #	Test Description	Tool Used	Expected Results	Pass/Fail
1	Successfully use command line interface start monitor	Bash/vim	Monitor start from command line interface, and record a password attempt in blacklist	Pass. Detailed results are attached.
2	Successfully use GUI start the monitor program	Bash	Monitor start by using GUI	Pass. Detailed results are attached.
3	Successfully use regular expression tester to test user defined pattern	Bash	Try regular expression tester many times to confirm user defined pattern works	Pass. Detailed results are attached.
SSH Service				
4	Successfully login SSH server	SSH client	The application won't match anything (command: #ssh server_ip)	Pass. Detailed results are attached.
5	Fail to input the correct password and matched by default method (awk)	SSH client	The application would match the record from log file, and add or update the record to the blacklist (command: #ssh server_ip)	Pass. Detailed results are attached.
6	Fail to input the correct password and matched by regular expression	SSH client	The application would match the record from log file, and add or update the record to the blacklist (command: #ssh server_ip)	Pass. Detailed results are attached.
7	More than one visitor fail to input the correct password	SSH client	The application would match the record from log file, and add the new record into the blacklist (command: #ssh server_ip)	Pass. Detailed results are attached.
8	One visitor try many times and then get blocked through iptables	SSH client & iptables	The application would match the record from log file, and delete the record from the blacklist The iptables block the visitor. (command: #ssh server_ip) (command: #iptables -L)	Pass. Detailed results are attached.

Rule #	Test Description	Tool Used	Expected Results	Pass/Fail
9	More than one visitors try many times and then get blocked through iptables	SSH client & iptables	The application would match the record from log file, and delete the record from the blacklist The iptables add a rules of blocking visitors. (command: #ssh server_ip) (command: #iptables -L)	Pass. Detailed results are attached.
10	Crontab task activate the application	crontab	The application would activate through crontab (* * * * * /app path/monitor.sh timelimit $target_ip $port)	Pass. Detailed results are attached.
11	After blocking period the visitor can access SSH service again	SSH client & iptables	After the blocking period, the blocking rule will be deleted, the visitor could access the ssh server again (command: #iptables -L) (command: #ssh server_ip)	Pass. Detailed results are attached.
FTP Service				
12	Successfully login FTP server	FTP client	The application won't match anything (command: #ftp server_ip)	Pass. Detailed results are attached.
13	Fail to input the correct password and matched by regular expression	FTP client	The application would match the record from log file, and add or update the record to the blacklist (command: #ftp server_ip)	Pass. Detailed results are attached.
14	Fail to input the correct password the 2nd time and matched by regular expression	FTP client	The application would match the record from log file, and add or update the record to the blacklist (command: #ftp server_ip)	Pass. Detailed results are attached.
15	More than one visitor fail to input the correct password	FTP client	The application would match the record from log file, and add the new record into the blacklist (command: #ftp server_ip)	Pass. Detailed results are attached.
16	One visitor try many times and then get blocked through iptables	FTP client& iptables	The application would match the record from log file, and delete the record from the blacklist The iptables add a rule of blocking visitor. (command: #iptables -L)	Pass. Detailed results are attached.

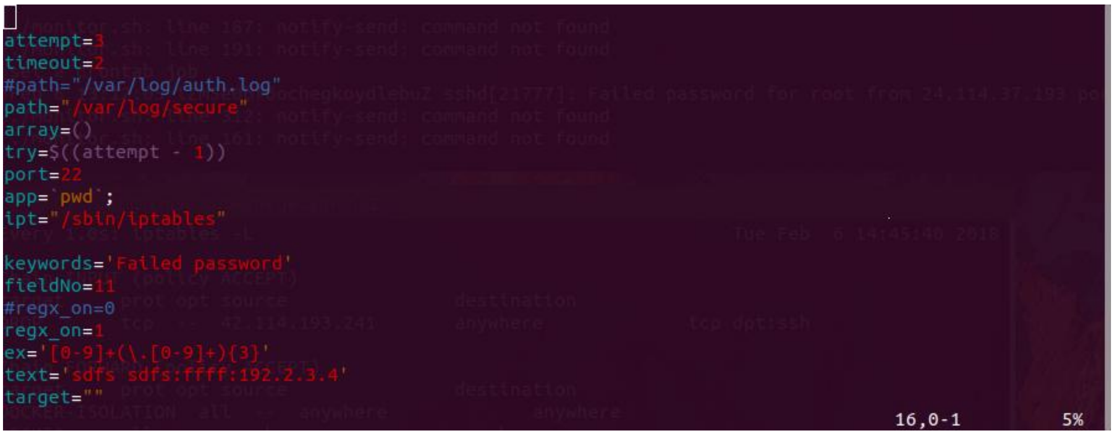
Rule #	Test Description	Tool Used	Expected Results	Pass/Fail
17	More than one visitors try many times and then get blocked through iptables	FTP client & iptables	The application would match the record from log file, and delete the record from the blacklist The iptables add a rules of blocking visitors. (command: #ftp server_ip) (command: #iptables -L)	Pass. Detailed results are attached.
18	Crontab task activate the application	crontab	The application would activate through crontab (grep CRON /var/log/syslog) (* * * * * /app path/monitor.sh timelimit $target_ip $port)	Pass. Detailed results are attached.
19	After blocking period the visitor can access FTP service again	FTP client& iptables	After the blocking period, the blocking rule will be deleted, the visitor could access the ssh server again (command: #iptables -L) (command: #ftp server_ip)	Pass. Detailed results are attached.

1.2 Test Case Descriptions

1.2.1 Test 1

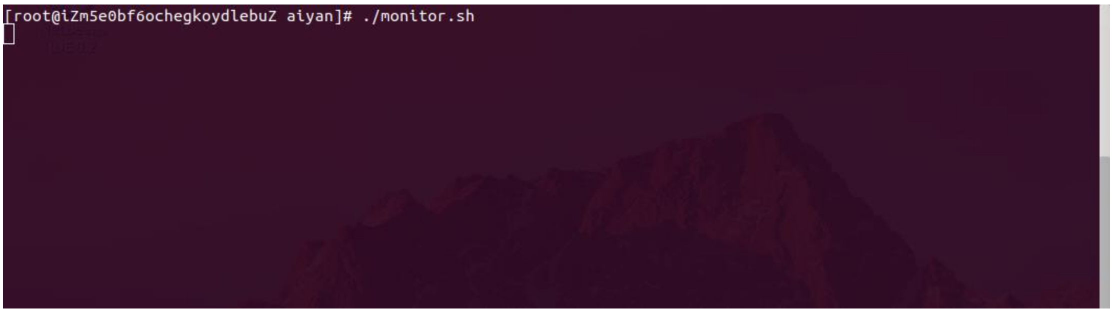
This was a simple test for how to use command line interface start monitor:

Set variables by hard code:



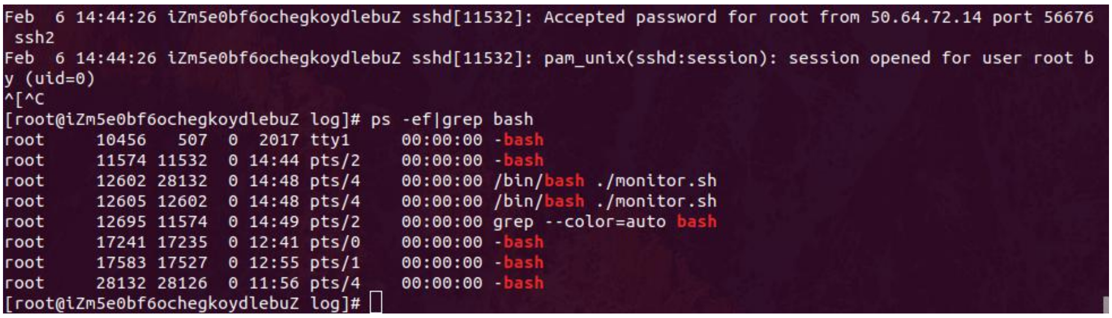
```
attempt=3
timeout=2
#path="/var/log/auth.log"
path="/var/log/secure"
array=()
try=$((attempt - 1))
port=22
app=`pwd`;
ipt="/sbin/iptables"

keywords='Failed password'
fieldNo=11
#regx_on=0
regx_on=1
ex='[0-9]+(\.[0-9]+){3}'
text='sdfs sdfs:ffff:192.2.3.4'
target=""
                                                              16,0-1         5%
```

Set general variable by command line:



```
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ./monitor.sh
```

Capture one line from log:



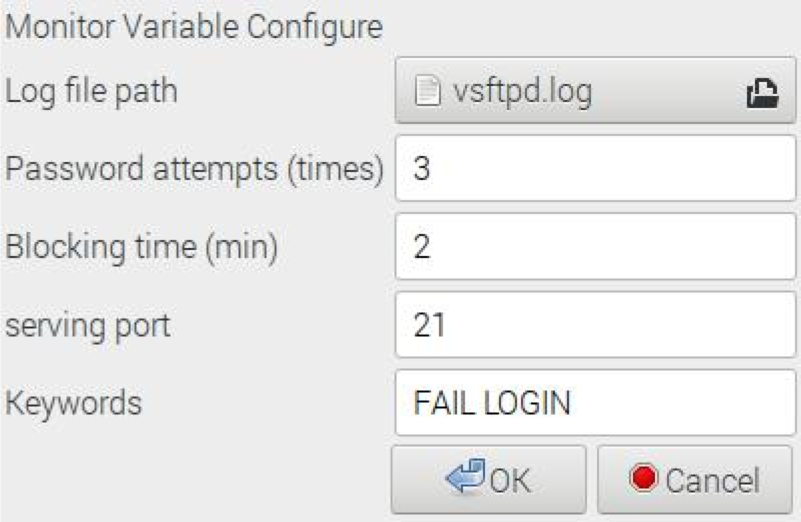
```
Feb  6 14:44:26 iZm5e0bf6ochegkoydlebuZ sshd[11532]: Accepted password for root from 50.64.72.14 port 56676
 ssh2
Feb  6 14:44:26 iZm5e0bf6ochegkoydlebuZ sshd[11532]: pam_unix(sshd:session): session opened for user root b
y (uid=0)
^[^C
[root@iZm5e0bf6ochegkoydlebuZ log]# ps -ef|grep bash
root     10456   507  0  2017 tty1     00:00:00 -bash
root     11574 11532  0 14:44 pts/2    00:00:00 -bash
root     12602 28132  0 14:48 pts/4    00:00:00 /bin/bash ./monitor.sh
root     12605 12602  0 14:48 pts/4    00:00:00 /bin/bash ./monitor.sh
root     12695 11574  0 14:49 pts/2    00:00:00 grep --color=auto bash
root     17241 17235  0 12:41 pts/0    00:00:00 -bash
root     17583 17527  0 12:55 pts/1    00:00:00 -bash
root     28132 28126  0 11:56 pts/4    00:00:00 -bash
[root@iZm5e0bf6ochegkoydlebuZ log]#
```

Test passes, we can see that the program is running at process no. 12602

1.2.2 Test 2

This was a simple test for how to use GUI run the monitor program.

Run the script: ./monitor.sh -gui



Monitor Variable Configure

Log file path: vsftpd.log

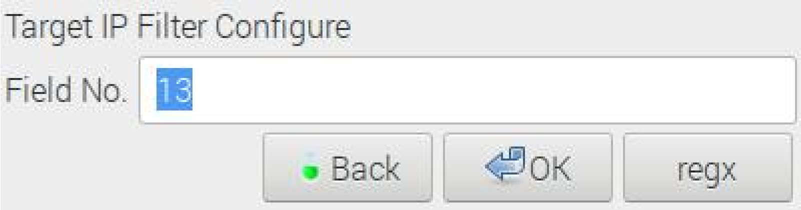
Password attempts (times): 3

Blocking time (min): 2

serving port: 21

Keywords: FAIL LOGIN

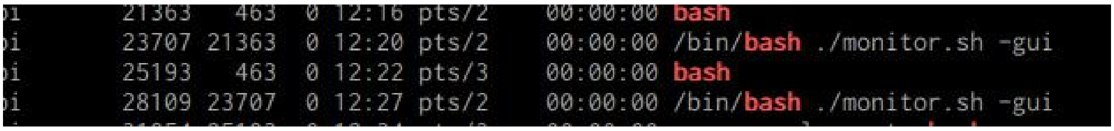
Set general variables then click OK button :



Target IP Filter Configure

Field No. 13

After settling variables then check: ps -ef | grep bash

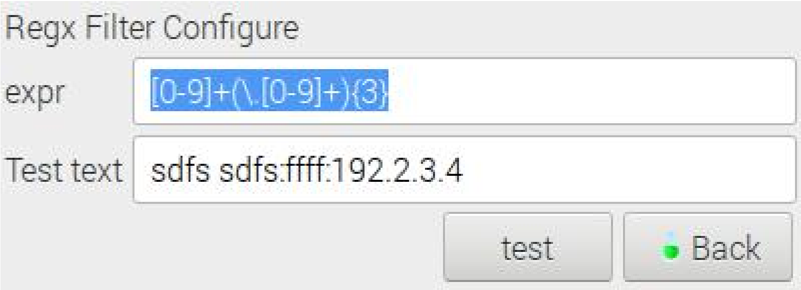


```
pi       21363   463  0 12:16 pts/2    00:00:00 bash
pi       23707 21363  0 12:20 pts/2    00:00:00 /bin/bash ./monitor.sh -gui
pi       25193   463  0 12:22 pts/3    00:00:00 bash
pi       28109 23707  0 12:27 pts/2    00:00:00 /bin/bash ./monitor.sh -gui
```

Test passes, we can see that the program is running at process no. 28109

1.2.3 Test 3

Use regular expression tester to test user defined pattern.
Run the script: ./monitor.sh -gui   step 3 after click regx button

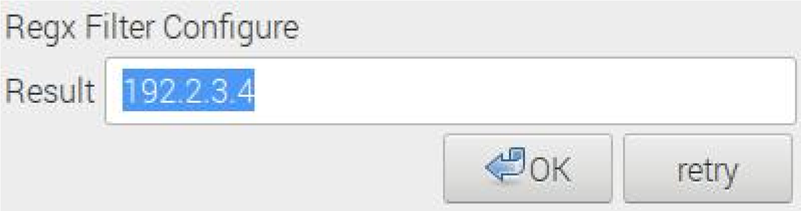


Regx Filter Configure

expr [0-9]+(\.[0-9]+){3}

Test text sdfs sdfs:ffff:192.2.3.4

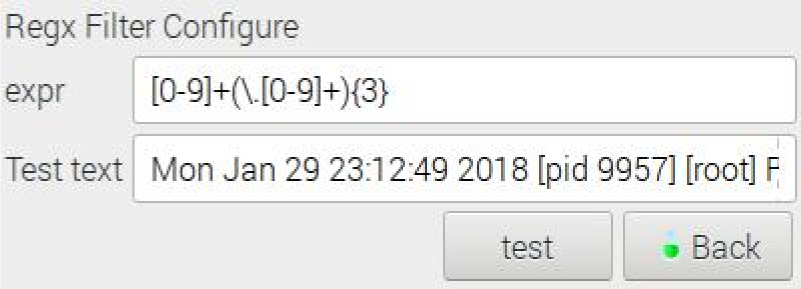
Test this pattern is works for the default testing text or not by click test button:



Regx Filter Configure

Result 192.2.3.4

Test succeed! Retry another one by click retry button:

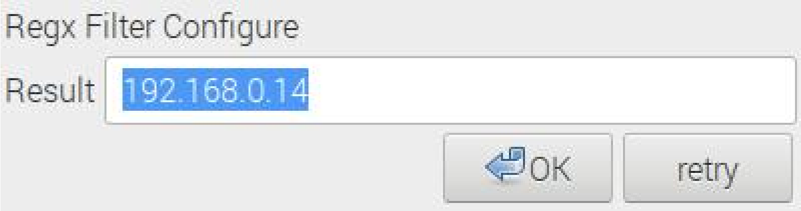


Regx Filter Configure

expr [0-9]+(\.[0-9]+){3}

Test text Mon Jan 29 23:12:49 2018 [pid 9957] [root] F

This time I try the line from vsftpd log:



Regx Filter Configure

Result 192.168.0.14

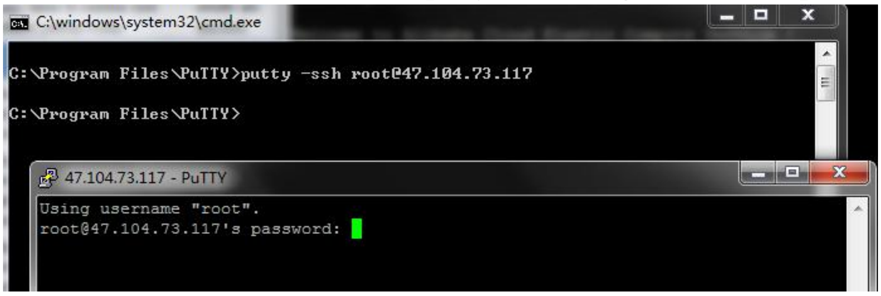
Test succeed!

Test passes.

1.2.4 Test 4

This test case is just test if the monitor program would take any action about visitor successfully login SSH server

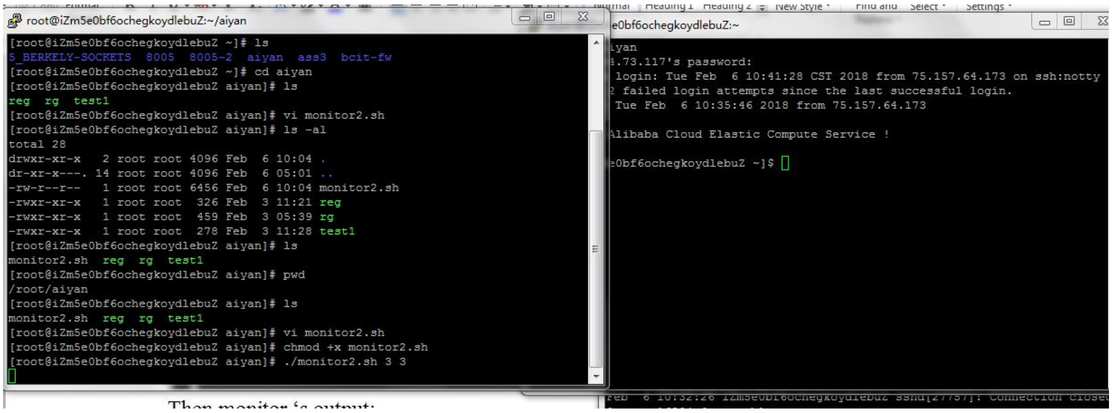
Run the command in client side: putty -ssh root@ssh_server_ip



```
C:\Program Files\PuTTY>putty -ssh root@47.104.73.117

C:\Program Files\PuTTY>
```

```
Using username "root".
root@47.104.73.117's password:
```

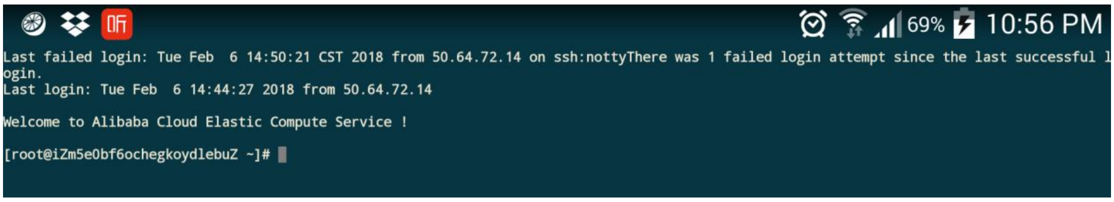
Then monitor ‘s output:



```
[root@iZm5e0bf6ochegkoydlebuZ ~]# ls
5_BERKELY-SOCKETS  8005  8005-2  aiyan  ass3  bcit-fw
[root@iZm5e0bf6ochegkoydlebuZ ~]# cd aiyan
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ls
reg  rg  test1
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# vi monitor2.sh
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ls -al
total 28
drwxr-xr-x   2 root root 4096 Feb  6 10:04 .
dr-xr-x---. 14 root root 4096 Feb  6 05:01 ..
-rw-r--r--   1 root root 6456 Feb  6 10:04 monitor2.sh
-rwxr-xr-x   1 root root  326 Feb  3 11:21 reg
-rwxr-xr-x   1 root root  459 Feb  3 05:39 rg
-rwxr-xr-x   1 root root  278 Feb  3 11:28 test1
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ls
monitor2.sh  reg  rg  test1
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# pwd
/root/aiyan
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ls
monitor2.sh  reg  rg  test1
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# vi monitor2.sh
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# chmod +x monitor2.sh
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ./monitor2.sh 3 3
```

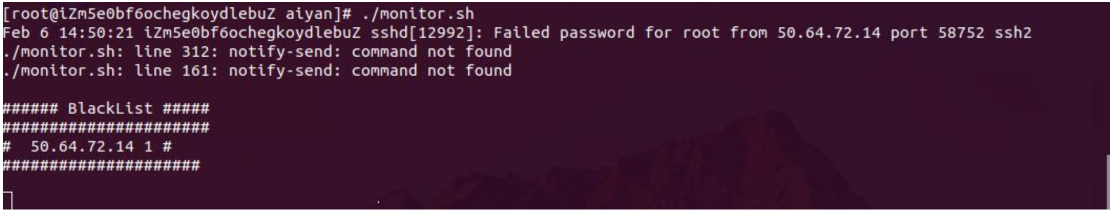
```
iyan
4.73.117's password:
 login: Tue Feb  6 10:41:28 CST 2018 from 75.157.64.173 on ssh:notty
2 failed login attempts since the last successful login.
 Tue Feb  6 10:35:46 2018 from 75.157.64.173

Alibaba Cloud Elastic Compute Service !

e0bf6ochegkoydlebuZ ~]$
```

[do twice by using different terminals or ip]



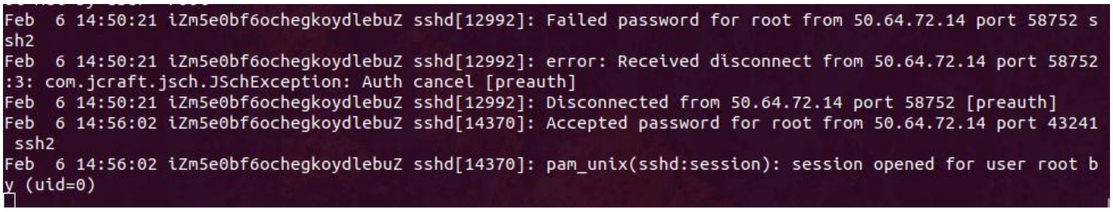
```
Last failed login: Tue Feb  6 14:50:21 CST 2018 from 50.64.72.14 on ssh:nottyThere was 1 failed login attempt since the last successful l
ogin.
Last login: Tue Feb  6 14:44:27 2018 from 50.64.72.14

Welcome to Alibaba Cloud Elastic Compute Service !

[root@iZm5e0bf6ochegkoydlebuZ ~]#
```

Monitor output:



```
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ./monitor.sh
Feb 6 14:50:21 iZm5e0bf6ochegkoydlebuZ sshd[12992]: Failed password for root from 50.64.72.14 port 58752 ssh2
./monitor.sh: line 312: notify-send: command not found
./monitor.sh: line 161: notify-send: command not found

###### BlackList #####
######################
#  50.64.72.14 1 #
#####################
```

Secure log shows



```
Feb  6 14:50:21 iZm5e0bf6ochegkoydlebuZ sshd[12992]: Failed password for root from 50.64.72.14 port 58752 s
sh2
Feb  6 14:50:21 iZm5e0bf6ochegkoydlebuZ sshd[12992]: error: Received disconnect from 50.64.72.14 port 58752
:3: com.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 14:50:21 iZm5e0bf6ochegkoydlebuZ sshd[12992]: Disconnected from 50.64.72.14 port 58752 [preauth]
Feb  6 14:56:02 iZm5e0bf6ochegkoydlebuZ sshd[14370]: Accepted password for root from 50.64.72.14 port 43241
 ssh2
Feb  6 14:56:02 iZm5e0bf6ochegkoydlebuZ sshd[14370]: pam_unix(sshd:session): session opened for user root b
y (uid=0)
```

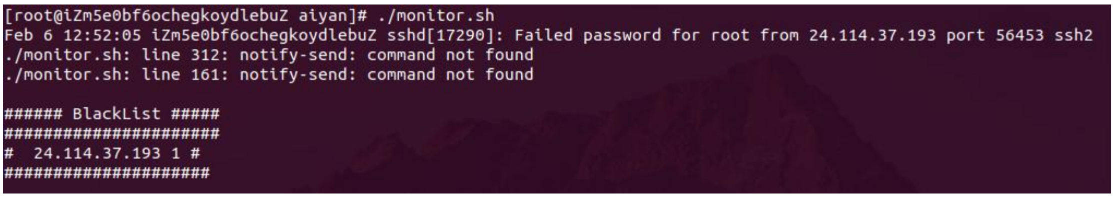
Test passed

1.2.5 Test 5

This test case test if one visitor fail to input the correct password for the first time then how the monitor program react when using default ip filter (awk).

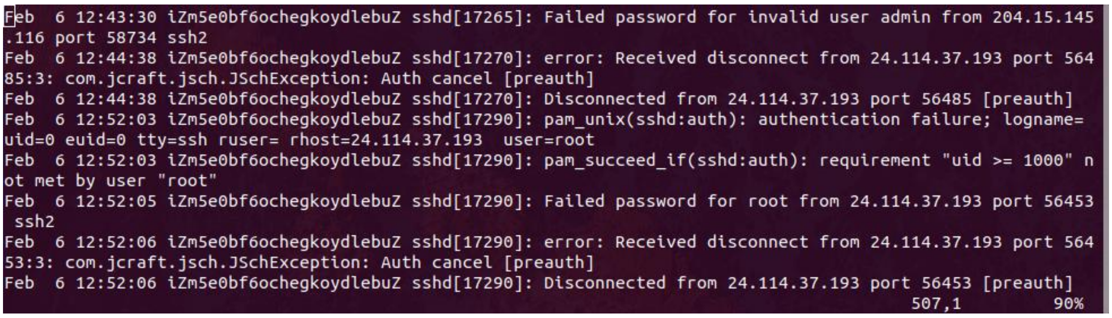
After changing the code of monitor.sh, re-test at Centos as
1) turn off the regex (check and make sure regx=0 by default);
2) sshing the host by 3 times wrongly passwd:

The server side monitor would output the current blacklist:



```
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ./monitor.sh
Feb 6 12:52:05 iZm5e0bf6ochegkoydlebuZ sshd[17290]: Failed password for root from 24.114.37.193 port 56453 ssh2
./monitor.sh: line 312: notify-send: command not found
./monitor.sh: line 161: notify-send: command not found

###### BlackList #####
######################
#  24.114.37.193 1 #
#####################
```

The record of log is:



```
Feb  6 12:43:30 iZm5e0bf6ochegkoydlebuZ sshd[17265]: Failed password for invalid user admin from 204.15.145
.116 port 58734 ssh2
Feb  6 12:44:38 iZm5e0bf6ochegkoydlebuZ sshd[17270]: error: Received disconnect from 24.114.37.193 port 564
85:3: com.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 12:44:38 iZm5e0bf6ochegkoydlebuZ sshd[17270]: Disconnected from 24.114.37.193 port 56485 [preauth]
Feb  6 12:52:03 iZm5e0bf6ochegkoydlebuZ sshd[17290]: pam_unix(sshd:auth): authentication failure; logname=
uid=0 euid=0 tty=ssh ruser= rhost=24.114.37.193  user=root
Feb  6 12:52:03 iZm5e0bf6ochegkoydlebuZ sshd[17290]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" n
ot met by user "root"
Feb  6 12:52:05 iZm5e0bf6ochegkoydlebuZ sshd[17290]: Failed password for root from 24.114.37.193 port 56453
 ssh2
Feb  6 12:52:06 iZm5e0bf6ochegkoydlebuZ sshd[17290]: error: Received disconnect from 24.114.37.193 port 564
53:3: com.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 12:52:06 iZm5e0bf6ochegkoydlebuZ sshd[17290]: Disconnected from 24.114.37.193 port 56453 [preauth]
                                                                                      507,1         90%
```

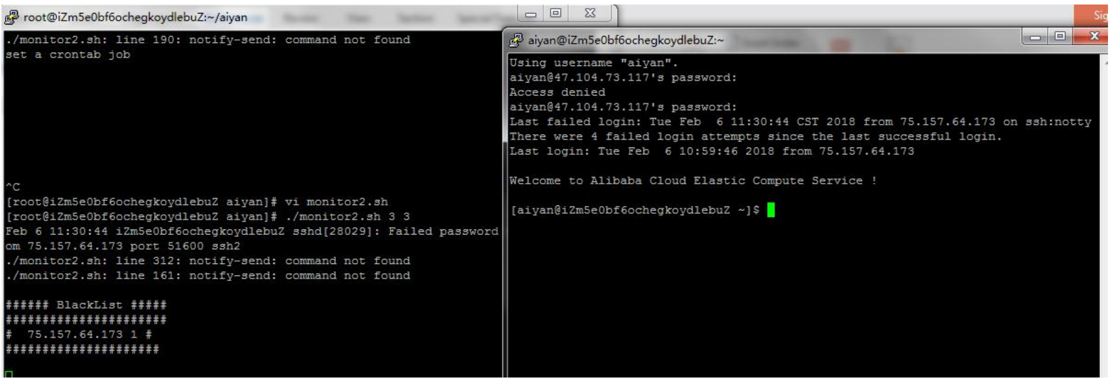
we can see that at 12:52 there is a time, client successfully logged in ssh server:

Test passed

1.2.6 Test 6

This test case test if one visitor fail to input the correct password then how the monitor program react and this time we are using regular expression to match keywords.

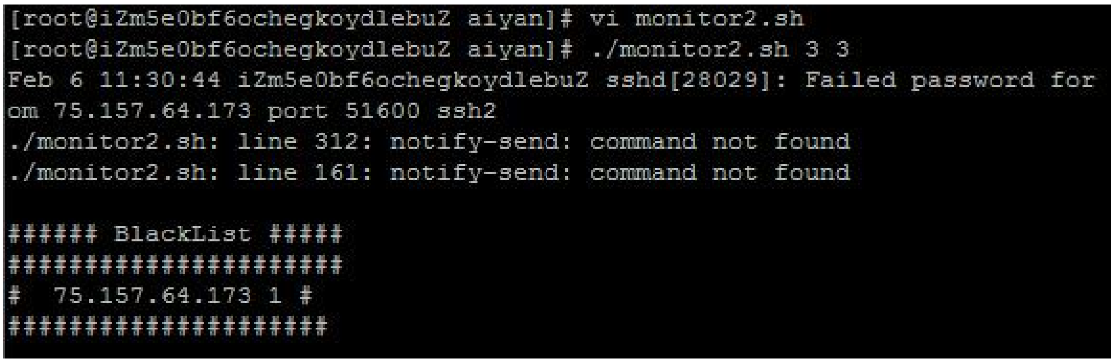
Set the regx=1 to turn on the regx
Ssh to the host by *putty aiyan@47.104.73.117*
One time wrong password, then input the correct password and can login


root@iZm5e0bf6ochegkoydlebuZ:~/aiyan
aiyan@iZm5e0bf6ochegkoydlebuZ:~

```
./monitor2.sh: line 190: notify-send: command not found
set a crontab job





^C
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# vi monitor2.sh
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ./monitor2.sh 3 3
Feb 6 11:30:44 iZm5e0bf6ochegkoydlebuZ sshd[28029]: Failed password
om 75.157.64.173 port 51600 ssh2
./monitor2.sh: line 312: notify-send: command not found
./monitor2.sh: line 161: notify-send: command not found

###### BlackList #####
######################
#  75.157.64.173 1 #
#####################
```

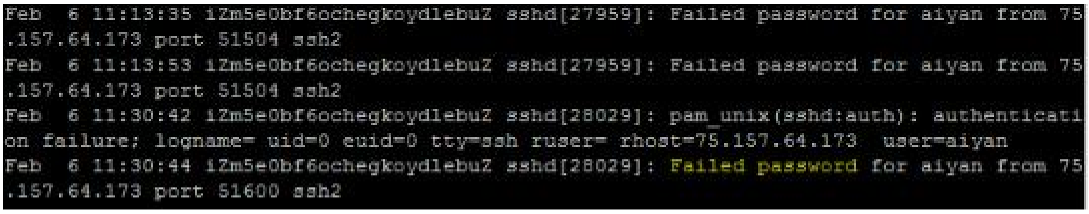
```
Using username "aiyan".
aiyan@47.104.73.117's password:
Access denied
aiyan@47.104.73.117's password:
Last failed login: Tue Feb  6 11:30:44 CST 2018 from 75.157.64.173 on ssh:notty
There were 4 failed login attempts since the last successful login.
Last login: Tue Feb  6 10:59:46 2018 from 75.157.64.173

Welcome to Alibaba Cloud Elastic Compute Service !

[aiyan@iZm5e0bf6ochegkoydlebuZ ~]$
```

Monitor output



```
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# vi monitor2.sh
[root@iZm5e0bf6ochegkoydlebuZ aiyan]# ./monitor2.sh 3 3
Feb  6 11:30:44 iZm5e0bf6ochegkoydlebuZ sshd[28029]: Failed password for
om 75.157.64.173 port 51600 ssh2
./monitor2.sh: line 312: notify-send: command not found
./monitor2.sh: line 161: notify-send: command not found

###### BlackList #####
######################
#  75.157.64.173 1 #
#####################
```

After trying for many times at least get the threshold attempt times by one visitor, the secure log:



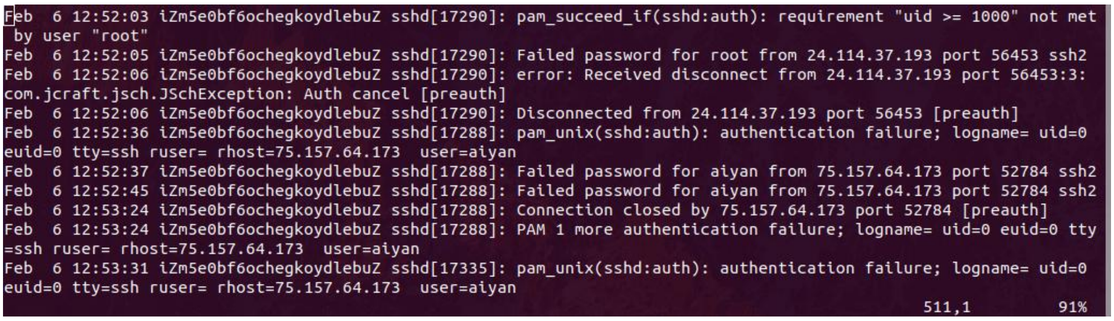
```
Feb  6 11:13:35 iZm5e0bf6ochegkoydlebuZ sshd[27959]: Failed password for aiyan from 75
.157.64.173 port 51504 ssh2
Feb  6 11:13:53 iZm5e0bf6ochegkoydlebuZ sshd[27959]: Failed password for aiyan from 75
.157.64.173 port 51504 ssh2
Feb  6 11:30:42 iZm5e0bf6ochegkoydlebuZ sshd[28029]: pam_unix(sshd:auth): authenticati
on failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=75.157.64.173  user=aiyan
Feb  6 11:30:44 iZm5e0bf6ochegkoydlebuZ sshd[28029]: Failed password for aiyan from 75
.157.64.173 port 51600 ssh2
```

Test passed.

1.2.7 Test 7

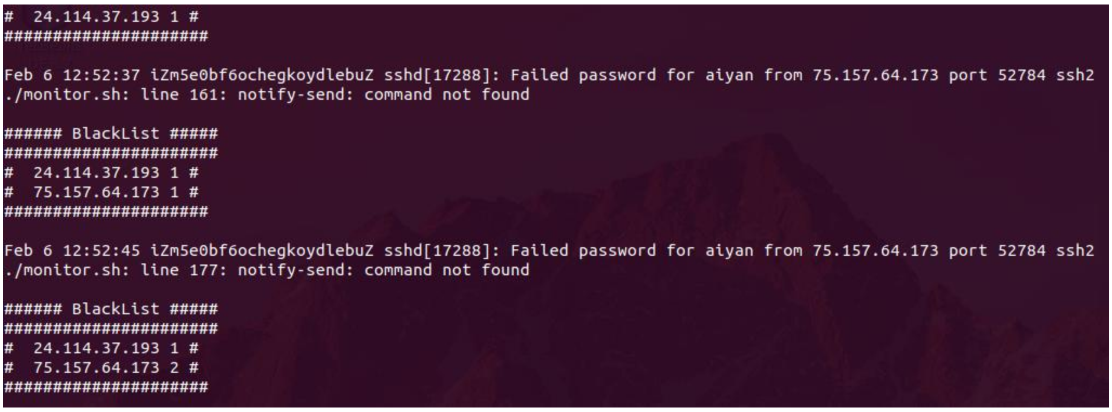
This test case test if more than one visitor fail to input the correct password what the monitor would respond for it.

After experience multiple visitor input wrong password the ssh log shows:



```
Feb  6 12:52:03 iZm5e0bf6ochegkoydlebuZ sshd[17290]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met
 by user "root"
Feb  6 12:52:05 iZm5e0bf6ochegkoydlebuZ sshd[17290]: Failed password for root from 24.114.37.193 port 56453 ssh2
Feb  6 12:52:06 iZm5e0bf6ochegkoydlebuZ sshd[17290]: error: Received disconnect from 24.114.37.193 port 56453:3:
com.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 12:52:06 iZm5e0bf6ochegkoydlebuZ sshd[17290]: Disconnected from 24.114.37.193 port 56453 [preauth]
Feb  6 12:52:36 iZm5e0bf6ochegkoydlebuZ sshd[17288]: pam_unix(sshd:auth): authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=75.157.64.173  user=aiyan
Feb  6 12:52:37 iZm5e0bf6ochegkoydlebuZ sshd[17288]: Failed password for aiyan from 75.157.64.173 port 52784 ssh2
Feb  6 12:52:45 iZm5e0bf6ochegkoydlebuZ sshd[17288]: Failed password for aiyan from 75.157.64.173 port 52784 ssh2
Feb  6 12:53:24 iZm5e0bf6ochegkoydlebuZ sshd[17288]: Connection closed by 75.157.64.173 port 52784 [preauth]
Feb  6 12:53:24 iZm5e0bf6ochegkoydlebuZ sshd[17288]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty
=ssh ruser= rhost=75.157.64.173  user=aiyan
Feb  6 12:53:31 iZm5e0bf6ochegkoydlebuZ sshd[17335]: pam_unix(sshd:auth): authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=75.157.64.173  user=aiyan
                                                                                          511,1          91%
```

And the monitor would output



```
#  24.114.37.193 1 #
#####################

Feb 6 12:52:37 iZm5e0bf6ochegkoydlebuZ sshd[17288]: Failed password for aiyan from 75.157.64.173 port 52784 ssh2
./monitor.sh: line 161: notify-send: command not found

###### BlackList #####
######################
#  24.114.37.193 1 #
#  75.157.64.173 1 #
#####################

Feb 6 12:52:45 iZm5e0bf6ochegkoydlebuZ sshd[17288]: Failed password for aiyan from 75.157.64.173 port 52784 ssh2
./monitor.sh: line 177: notify-send: command not found

###### BlackList #####
######################
#  24.114.37.193 1 #
#  75.157.64.173 2 #
#####################
```

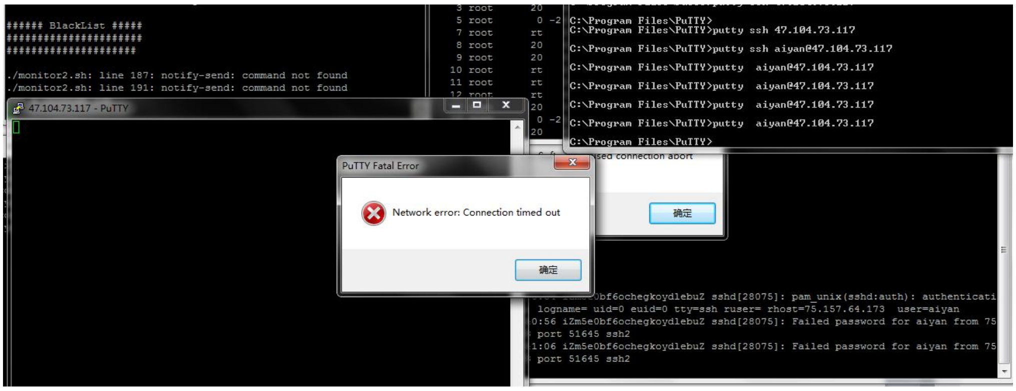
We can find that there are two ip in the current blacklist, and one of them try the password twice.

Test passed.

1.2.8 Test 8

This test case test if one visitor try many times that over the threshold then what the monitor would respond for it.

After 3 times wrong password, the connection is reset and no more new connection is allowed.
Functional Pass:



```
###### BlackList #####
######################
#####################

./monitor2.sh: line 187: notify-send: command not found
./monitor2.sh: line 191: notify-send: command not found
```

```
  3 root      20
  5 root       0 -2
  7 root      rt
  8 root      20
  9 root      20
 10 root      rt
 11 root      rt
 12 root      rt
```

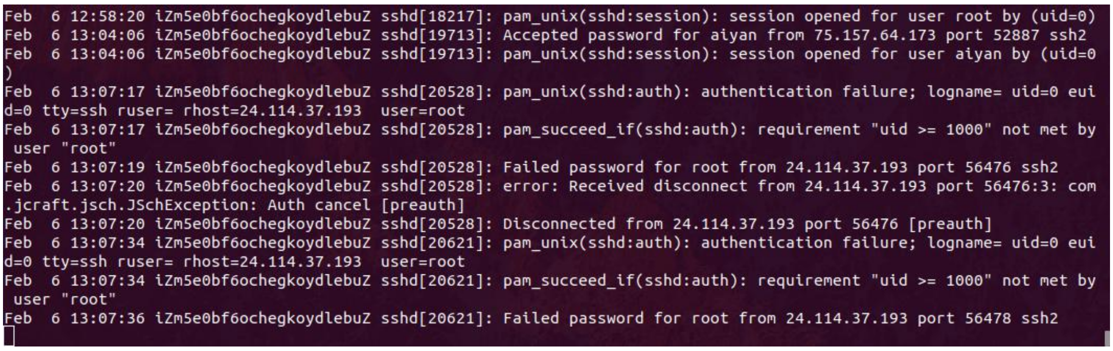
```
C:\Program Files\PuTTY>
C:\Program Files\PuTTY>putty ssh 47.104.73.117

C:\Program Files\PuTTY>putty ssh aiyan@47.104.73.117

C:\Program Files\PuTTY>putty  aiyan@47.104.73.117

C:\Program Files\PuTTY>putty  aiyan@47.104.73.117

C:\Program Files\PuTTY>putty  aiyan@47.104.73.117

C:\Program Files\PuTTY>putty  aiyan@47.104.73.117

C:\Program Files\PuTTY>
```

PuTTY Fatal Error

Network error: Connection timed out

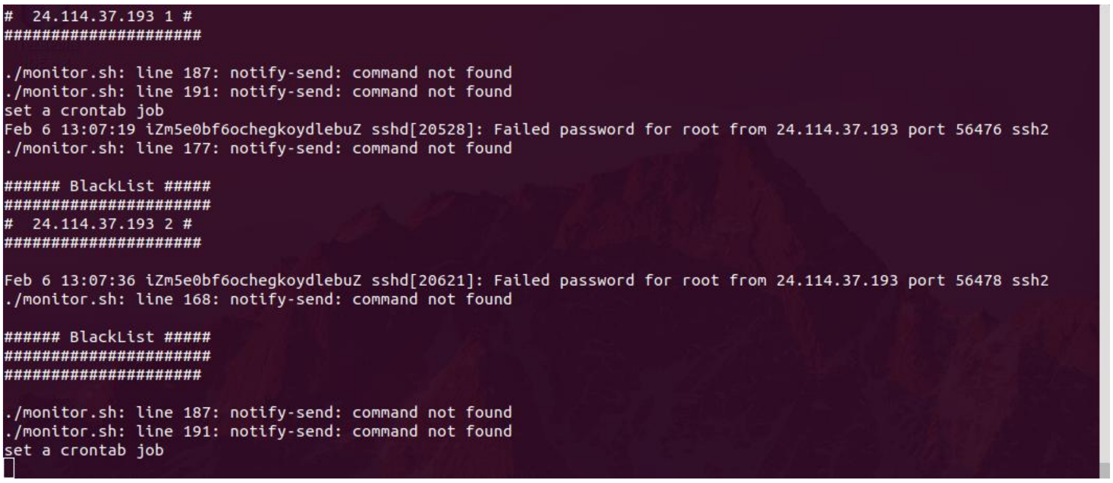
```
0bf6ochegkoydlebuZ sshd[28075]: pam_unix(sshd:auth): authenticati
 logname= uid=0 euid=0 tty=ssh ruser= rhost=75.157.64.173  user=aiyan
0:56 iZm5e0bf6ochegkoydlebuZ sshd[28075]: Failed password for aiyan from 75
 port 51645 ssh2
1:06 iZm5e0bf6ochegkoydlebuZ sshd[28075]: Failed password for aiyan from 75
 port 51645 ssh2
```

After trying for many times at least get the threshold attempt times by one, the secure log:



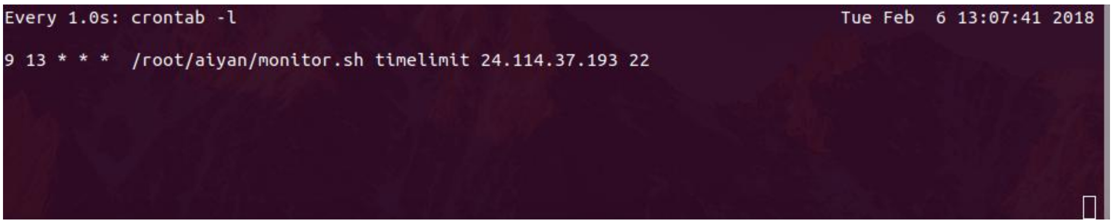
```
Feb  6 12:58:20 iZm5e0bf6ochegkoydlebuZ sshd[18217]: pam_unix(sshd:session): session opened for user root by (uid=0)
Feb  6 13:04:06 iZm5e0bf6ochegkoydlebuZ sshd[19713]: Accepted password for aiyan from 75.157.64.173 port 52887 ssh2
Feb  6 13:04:06 iZm5e0bf6ochegkoydlebuZ sshd[19713]: pam_unix(sshd:session): session opened for user aiyan by (uid=0
)
Feb  6 13:07:17 iZm5e0bf6ochegkoydlebuZ sshd[20528]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
d=0 tty=ssh ruser= rhost=24.114.37.193  user=root
Feb  6 13:07:17 iZm5e0bf6ochegkoydlebuZ sshd[20528]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
 user "root"
Feb  6 13:07:19 iZm5e0bf6ochegkoydlebuZ sshd[20528]: Failed password for root from 24.114.37.193 port 56476 ssh2
Feb  6 13:07:20 iZm5e0bf6ochegkoydlebuZ sshd[20528]: error: Received disconnect from 24.114.37.193 port 56476:3: com
.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 13:07:20 iZm5e0bf6ochegkoydlebuZ sshd[20528]: Disconnected from 24.114.37.193 port 56476 [preauth]
Feb  6 13:07:34 iZm5e0bf6ochegkoydlebuZ sshd[20621]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
d=0 tty=ssh ruser= rhost=24.114.37.193  user=root
Feb  6 13:07:34 iZm5e0bf6ochegkoydlebuZ sshd[20621]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
 user "root"
Feb  6 13:07:36 iZm5e0bf6ochegkoydlebuZ sshd[20621]: Failed password for root from 24.114.37.193 port 56478 ssh2
```

Monitor output

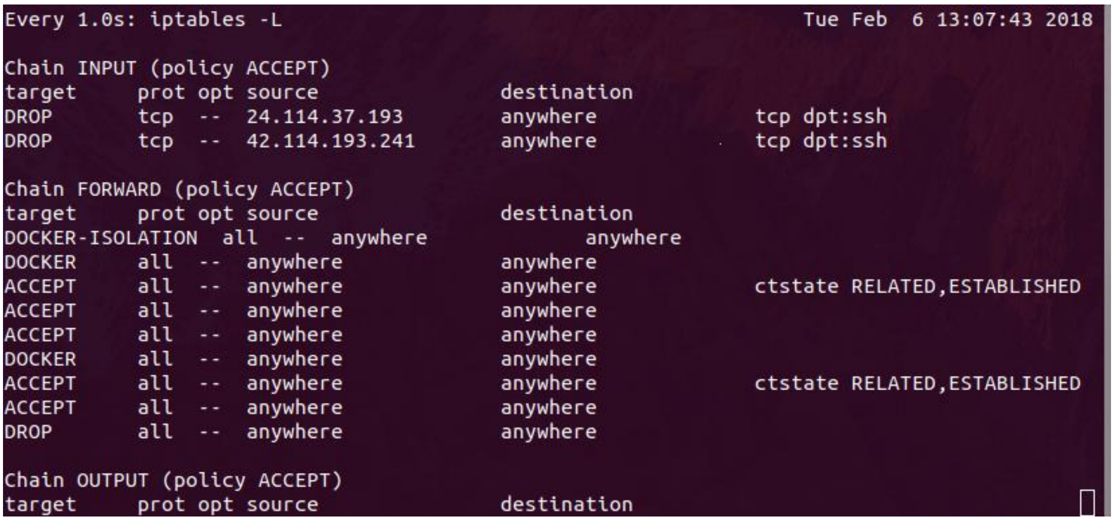


```
#  24.114.37.193 1 #
#####################

./monitor.sh: line 187: notify-send: command not found
./monitor.sh: line 191: notify-send: command not found
set a crontab job
Feb 6 13:07:19 iZm5e0bf6ochegkoydlebuZ sshd[20528]: Failed password for root from 24.114.37.193 port 56476 ssh2
./monitor.sh: line 177: notify-send: command not found

###### BlackList #####
######################
#  24.114.37.193 2 #
#####################

Feb 6 13:07:36 iZm5e0bf6ochegkoydlebuZ sshd[20621]: Failed password for root from 24.114.37.193 port 56478 ssh2
./monitor.sh: line 168: notify-send: command not found

###### BlackList #####
######################
#####################

./monitor.sh: line 187: notify-send: command not found
./monitor.sh: line 191: notify-send: command not found
set a crontab job
```

We can find that the blacklist is cleared;
at the same time check crontab and iptables rule:



```
Every 1.0s: crontab -l                                              Tue Feb  6 13:07:41 2018

9 13 * * *  /root/aiyan/monitor.sh timelimit 24.114.37.193 22
```



```
Every 1.0s: iptables -L                                     Tue Feb  6 13:07:43 2018

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  24.114.37.193        anywhere             tcp dpt:ssh
DROP       tcp  --  42.114.193.241       anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DOCKER-ISOLATION  all  --  anywhere             anywhere
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
DROP       all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

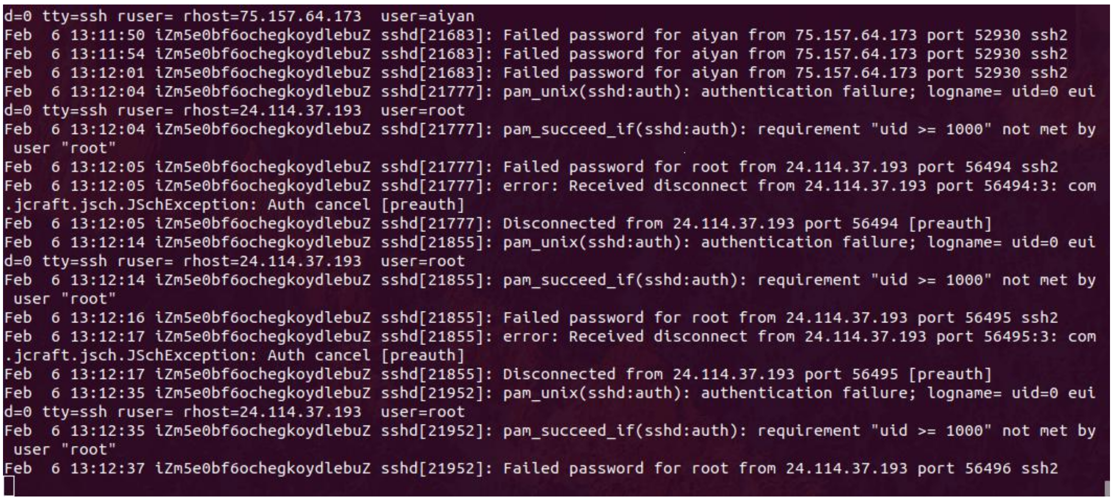
So we can see that the monitor add a firewall blocking rule and a crontab task in this process.

Test passed.

1.2.9 Test 9

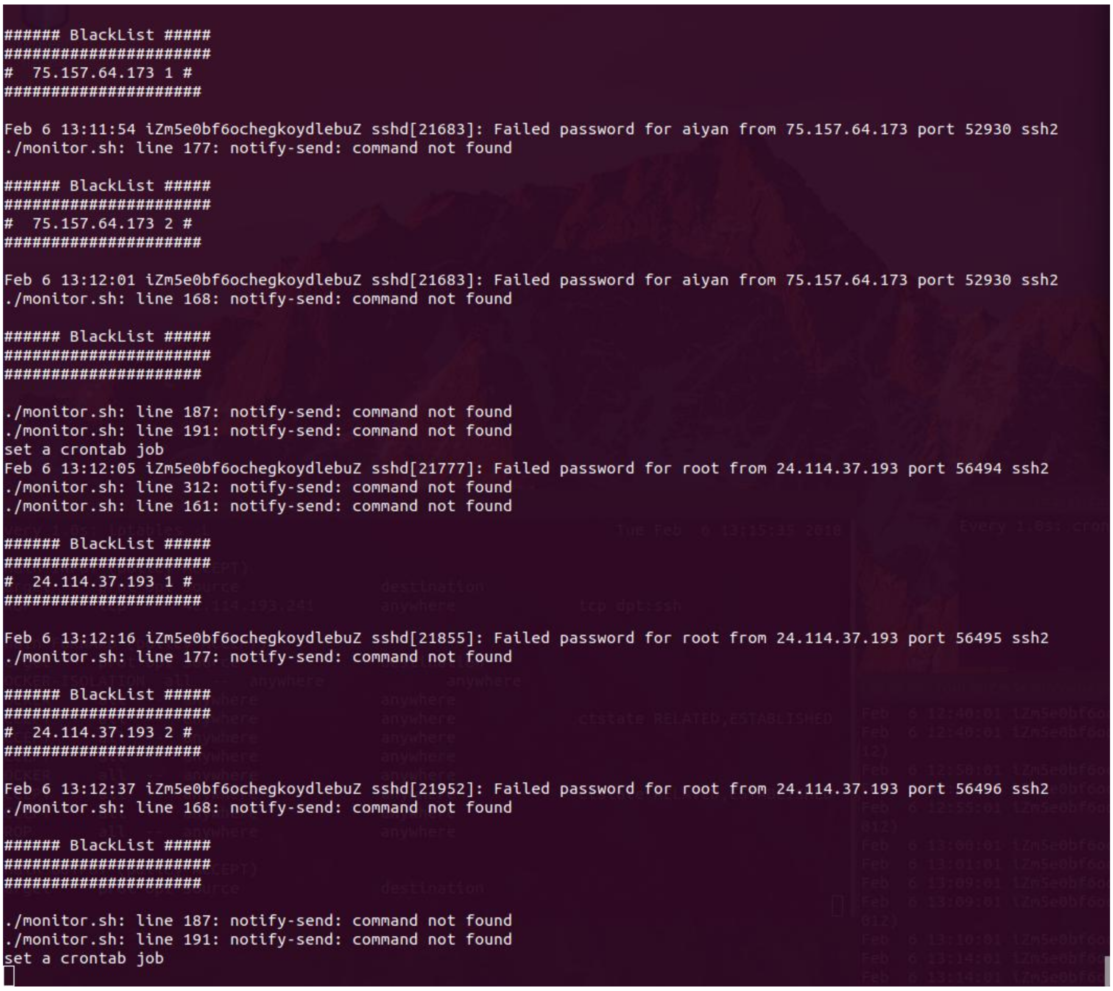
This test case test if more than one visitors try many times that over the threshold then what the monitor would respond for it.

After trying for many times at least get the threshold attempt times by multiple visitors, the secure log:



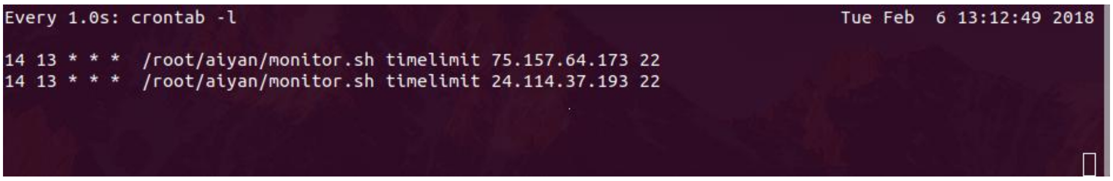
```
d=0 tty=ssh ruser= rhost=75.157.64.173  user=aiyan
Feb  6 13:11:50 iZm5e0bf6ochegkoydlebuZ sshd[21683]: Failed password for aiyan from 75.157.64.173 port 52930 ssh2
Feb  6 13:11:54 iZm5e0bf6ochegkoydlebuZ sshd[21683]: Failed password for aiyan from 75.157.64.173 port 52930 ssh2
Feb  6 13:12:01 iZm5e0bf6ochegkoydlebuZ sshd[21683]: Failed password for aiyan from 75.157.64.173 port 52930 ssh2
Feb  6 13:12:04 iZm5e0bf6ochegkoydlebuZ sshd[21777]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
d=0 tty=ssh ruser= rhost=24.114.37.193  user=root
Feb  6 13:12:04 iZm5e0bf6ochegkoydlebuZ sshd[21777]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
 user "root"
Feb  6 13:12:05 iZm5e0bf6ochegkoydlebuZ sshd[21777]: Failed password for root from 24.114.37.193 port 56494 ssh2
Feb  6 13:12:05 iZm5e0bf6ochegkoydlebuZ sshd[21777]: error: Received disconnect from 24.114.37.193 port 56494:3: com
.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 13:12:05 iZm5e0bf6ochegkoydlebuZ sshd[21777]: Disconnected from 24.114.37.193 port 56494 [preauth]
Feb  6 13:12:14 iZm5e0bf6ochegkoydlebuZ sshd[21855]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
d=0 tty=ssh ruser= rhost=24.114.37.193  user=root
Feb  6 13:12:14 iZm5e0bf6ochegkoydlebuZ sshd[21855]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
 user "root"
Feb  6 13:12:16 iZm5e0bf6ochegkoydlebuZ sshd[21855]: Failed password for root from 24.114.37.193 port 56495 ssh2
Feb  6 13:12:17 iZm5e0bf6ochegkoydlebuZ sshd[21855]: error: Received disconnect from 24.114.37.193 port 56495:3: com
.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 13:12:17 iZm5e0bf6ochegkoydlebuZ sshd[21855]: Disconnected from 24.114.37.193 port 56495 [preauth]
Feb  6 13:12:35 iZm5e0bf6ochegkoydlebuZ sshd[21952]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
d=0 tty=ssh ruser= rhost=24.114.37.193  user=root
Feb  6 13:12:35 iZm5e0bf6ochegkoydlebuZ sshd[21952]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
 user "root"
Feb  6 13:12:37 iZm5e0bf6ochegkoydlebuZ sshd[21952]: Failed password for root from 24.114.37.193 port 56496 ssh2
```

Monitor output

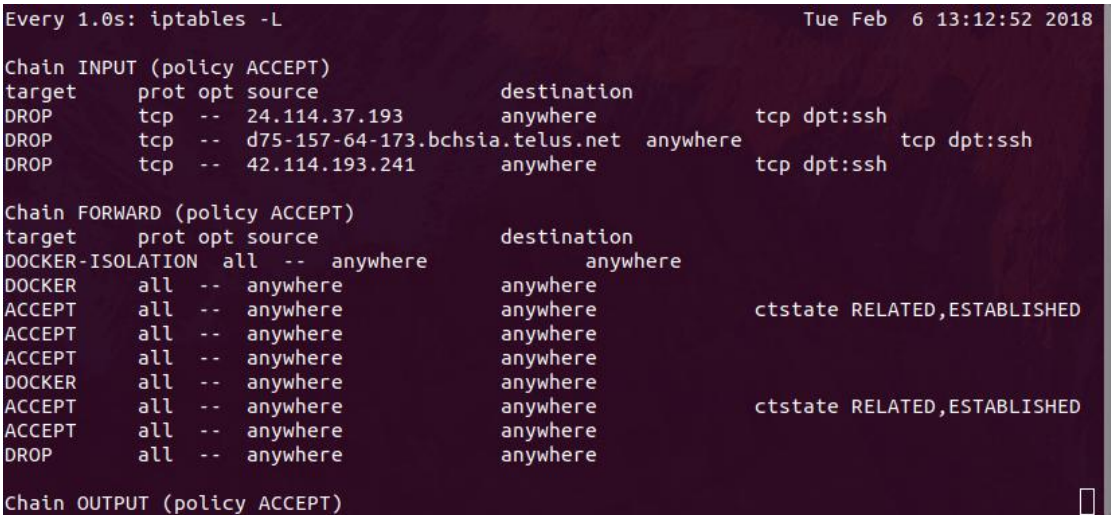


```
###### BlackList #####
######################
#  75.157.64.173 1 #
#####################

Feb 6 13:11:54 iZm5e0bf6ochegkoydlebuZ sshd[21683]: Failed password for aiyan from 75.157.64.173 port 52930 ssh2
./monitor.sh: line 177: notify-send: command not found

###### BlackList #####
######################
#  75.157.64.173 2 #
#####################

Feb 6 13:12:01 iZm5e0bf6ochegkoydlebuZ sshd[21683]: Failed password for aiyan from 75.157.64.173 port 52930 ssh2
./monitor.sh: line 168: notify-send: command not found

###### BlackList #####
######################
#####################

./monitor.sh: line 187: notify-send: command not found
./monitor.sh: line 191: notify-send: command not found
set a crontab job
Feb 6 13:12:05 iZm5e0bf6ochegkoydlebuZ sshd[21777]: Failed password for root from 24.114.37.193 port 56494 ssh2
./monitor.sh: line 312: notify-send: command not found
./monitor.sh: line 161: notify-send: command not found

###### BlackList #####
######################
#  24.114.37.193 1 #
#####################

Feb 6 13:12:16 iZm5e0bf6ochegkoydlebuZ sshd[21855]: Failed password for root from 24.114.37.193 port 56495 ssh2
./monitor.sh: line 177: notify-send: command not found

###### BlackList #####
######################
#  24.114.37.193 2 #
#####################

Feb 6 13:12:37 iZm5e0bf6ochegkoydlebuZ sshd[21952]: Failed password for root from 24.114.37.193 port 56496 ssh2
./monitor.sh: line 168: notify-send: command not found

###### BlackList #####
######################
#####################

./monitor.sh: line 187: notify-send: command not found
./monitor.sh: line 191: notify-send: command not found
set a crontab job
```

We can find that the blacklist is cleared;
at the same time check crontab and iptables rule:



```
Every 1.0s: crontab -l                                         Tue Feb  6 13:12:49 2018

14 13 * * *  /root/aiyan/monitor.sh timelimit 75.157.64.173 22
14 13 * * *  /root/aiyan/monitor.sh timelimit 24.114.37.193 22
```



```
Every 1.0s: iptables -L                                         Tue Feb  6 13:12:52 2018

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  24.114.37.193        anywhere             tcp dpt:ssh
DROP       tcp  --  d75-157-64-173.bchsia.telus.net  anywhere             tcp dpt:ssh
DROP       tcp  --  42.114.193.241       anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DOCKER-ISOLATION  all  --  anywhere             anywhere
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
DROP       all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
```

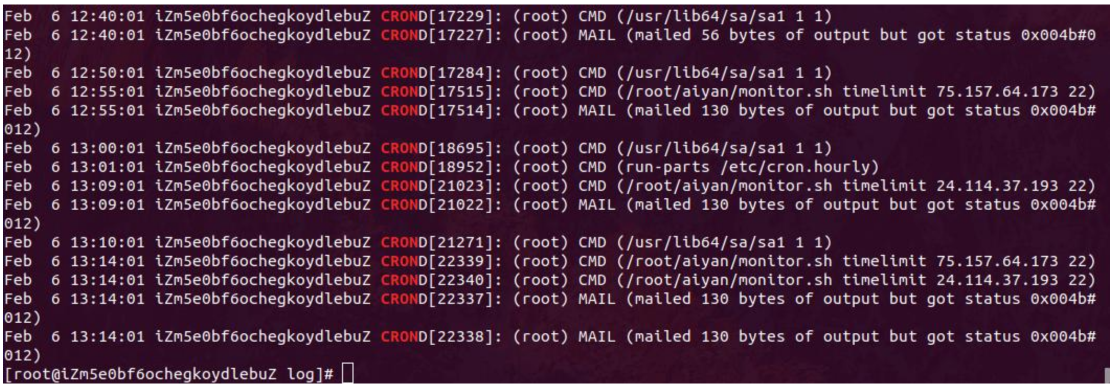
So we can see that the monitor add a firewall blocking rule and a crontab task in this process for both ip.

Test passed.

1.2.10 Test 10

This test case test how crontab task activate the application after monitor insert a rule into crontab.

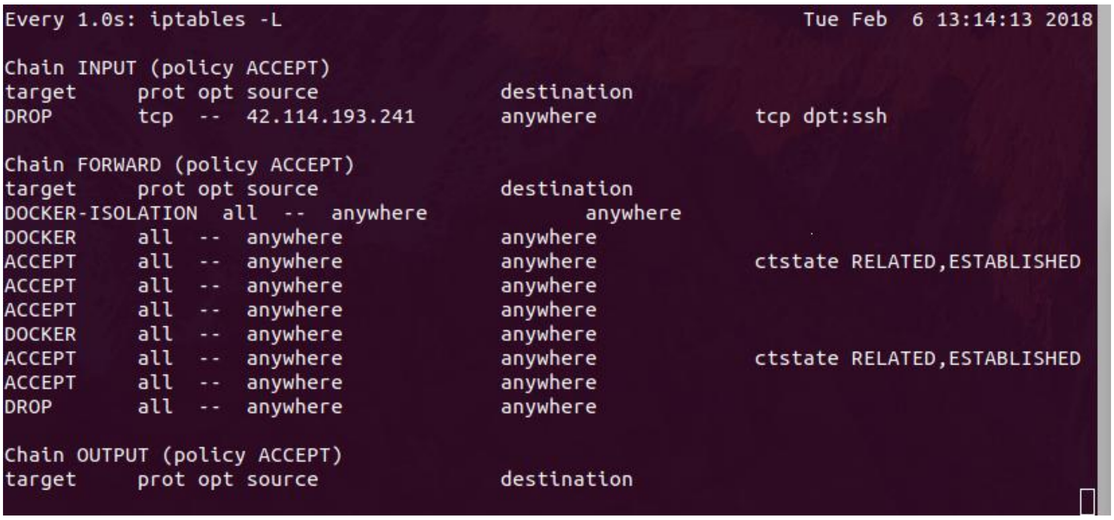
When the task start time pass, the crontab log record:
Command: grep CRON /var/log/syslog



```
Feb  6 12:40:01 iZm5e0bf6ochegkoydlebuZ CROND[17229]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Feb  6 12:40:01 iZm5e0bf6ochegkoydlebuZ CROND[17227]: (root) MAIL (mailed 56 bytes of output but got status 0x004b#0
12)
Feb  6 12:50:01 iZm5e0bf6ochegkoydlebuZ CROND[17284]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Feb  6 12:55:01 iZm5e0bf6ochegkoydlebuZ CROND[17515]: (root) CMD (/root/aiyan/monitor.sh timelimit 75.157.64.173 22)
Feb  6 12:55:01 iZm5e0bf6ochegkoydlebuZ CROND[17514]: (root) MAIL (mailed 130 bytes of output but got status 0x004b#
012)
Feb  6 13:00:01 iZm5e0bf6ochegkoydlebuZ CROND[18695]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Feb  6 13:01:01 iZm5e0bf6ochegkoydlebuZ CROND[18952]: (root) CMD (run-parts /etc/cron.hourly)
Feb  6 13:09:01 iZm5e0bf6ochegkoydlebuZ CROND[21023]: (root) CMD (/root/aiyan/monitor.sh timelimit 24.114.37.193 22)
Feb  6 13:09:01 iZm5e0bf6ochegkoydlebuZ CROND[21022]: (root) MAIL (mailed 130 bytes of output but got status 0x004b#
012)
Feb  6 13:10:01 iZm5e0bf6ochegkoydlebuZ CROND[21271]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Feb  6 13:14:01 iZm5e0bf6ochegkoydlebuZ CROND[22339]: (root) CMD (/root/aiyan/monitor.sh timelimit 75.157.64.173 22)
Feb  6 13:14:01 iZm5e0bf6ochegkoydlebuZ CROND[22340]: (root) CMD (/root/aiyan/monitor.sh timelimit 24.114.37.193 22)
Feb  6 13:14:01 iZm5e0bf6ochegkoydlebuZ CROND[22337]: (root) MAIL (mailed 130 bytes of output but got status 0x004b#
012)
Feb  6 13:14:01 iZm5e0bf6ochegkoydlebuZ CROND[22338]: (root) MAIL (mailed 130 bytes of output but got status 0x004b#
012)
[root@iZm5e0bf6ochegkoydlebuZ log]# 
```

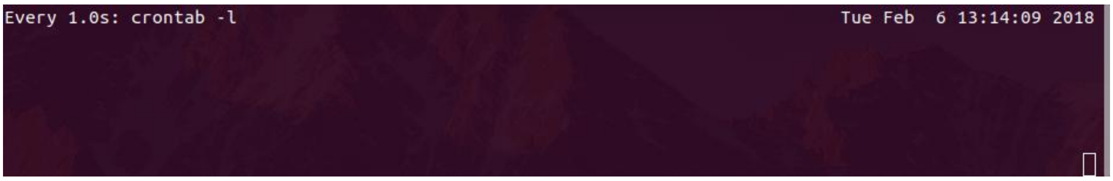
This cron task would clear the firewall rule:

After running the task, this task would be clear from the crontab:



```
Every 1.0s: iptables -L                                 Tue Feb  6 13:14:13 2018

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  42.114.193.241       anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DOCKER-ISOLATION  all  --  anywhere             anywhere
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
DROP       all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

And then clear itself from crontab:



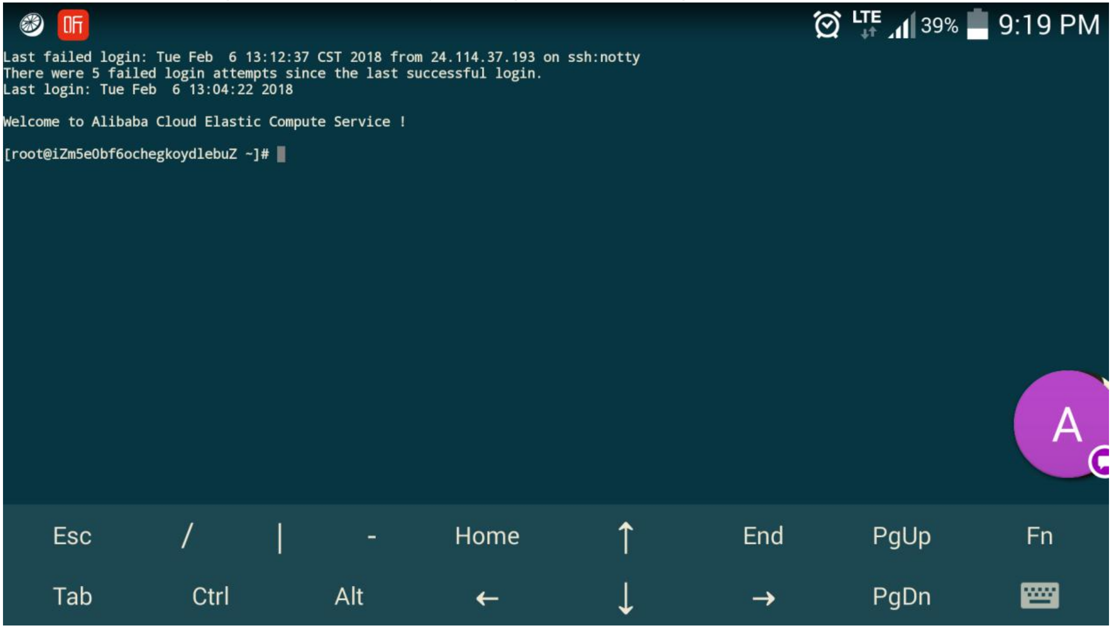
```
Every 1.0s: crontab -l                                  Tue Feb  6 13:14:09 2018
```

Test passed.

1.2.11 Test 11

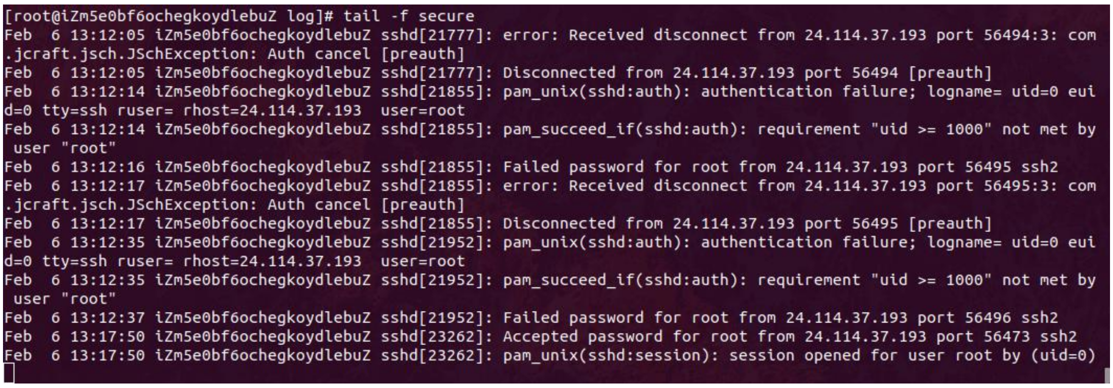
This test case test if after blocking period the visitor can access SSH service again

After the blocking time, visitor try to log in the ssh again:



```
Last failed login: Tue Feb  6 13:12:37 CST 2018 from 24.114.37.193 on ssh:notty
There were 5 failed login attempts since the last successful login.
Last login: Tue Feb  6 13:04:22 2018

Welcome to Alibaba Cloud Elastic Compute Service !

[root@iZm5e0bf6ochegkoydlebuZ ~]# 
```

And the secure log shows:



```
[root@iZm5e0bf6ochegkoydlebuZ log]# tail -f secure
Feb  6 13:12:05 iZm5e0bf6ochegkoydlebuZ sshd[21777]: error: Received disconnect from 24.114.37.193 port 56494:3: com
.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 13:12:05 iZm5e0bf6ochegkoydlebuZ sshd[21777]: Disconnected from 24.114.37.193 port 56494 [preauth]
Feb  6 13:12:14 iZm5e0bf6ochegkoydlebuZ sshd[21855]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
d=0 tty=ssh ruser= rhost=24.114.37.193  user=root
Feb  6 13:12:14 iZm5e0bf6ochegkoydlebuZ sshd[21855]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
 user "root"
Feb  6 13:12:16 iZm5e0bf6ochegkoydlebuZ sshd[21855]: Failed password for root from 24.114.37.193 port 56495 ssh2
Feb  6 13:12:17 iZm5e0bf6ochegkoydlebuZ sshd[21855]: error: Received disconnect from 24.114.37.193 port 56495:3: com
.jcraft.jsch.JSchException: Auth cancel [preauth]
Feb  6 13:12:17 iZm5e0bf6ochegkoydlebuZ sshd[21855]: Disconnected from 24.114.37.193 port 56495 [preauth]
Feb  6 13:12:35 iZm5e0bf6ochegkoydlebuZ sshd[21952]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
d=0 tty=ssh ruser= rhost=24.114.37.193  user=root
Feb  6 13:12:35 iZm5e0bf6ochegkoydlebuZ sshd[21952]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
 user "root"
Feb  6 13:12:37 iZm5e0bf6ochegkoydlebuZ sshd[21952]: Failed password for root from 24.114.37.193 port 56496 ssh2
Feb  6 13:17:50 iZm5e0bf6ochegkoydlebuZ sshd[23262]: Accepted password for root from 24.114.37.193 port 56473 ssh2
Feb  6 13:17:50 iZm5e0bf6ochegkoydlebuZ sshd[23262]: pam_unix(sshd:session): session opened for user root by (uid=0)
```

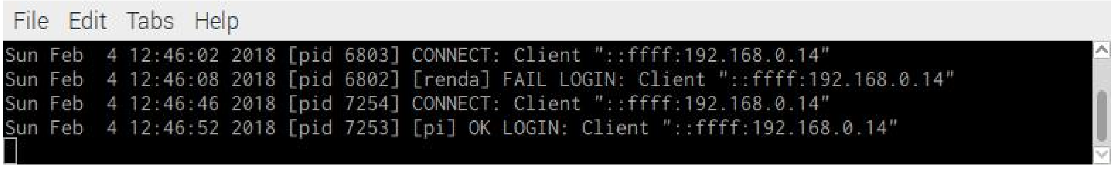
Accessed successfully

Test passed.

1.2.12 Test 12

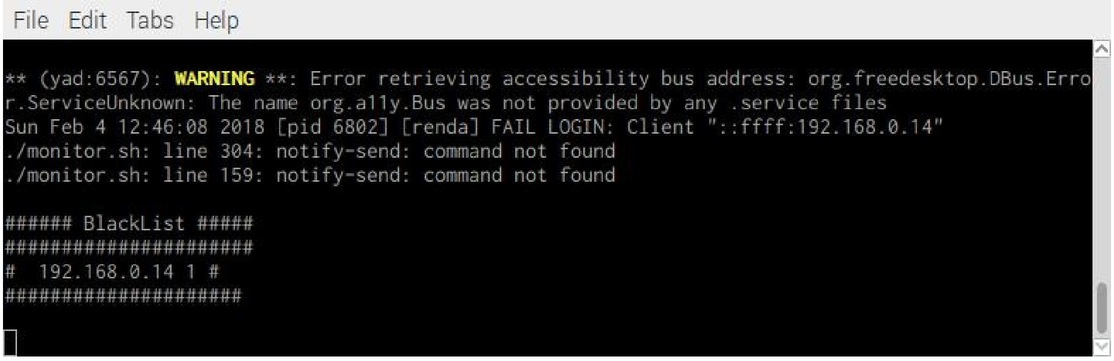
This test case is just test if the monitor program would take any action about visitor successfully login FTP server

Run the command in client side: ftp $ftp_server_ip



```
Sun Feb  4 12:46:02 2018 [pid 6803] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:46:08 2018 [pid 6802] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 12:46:46 2018 [pid 7254] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:46:52 2018 [pid 7253] [pi] OK LOGIN: Client "::ffff:192.168.0.14"
```

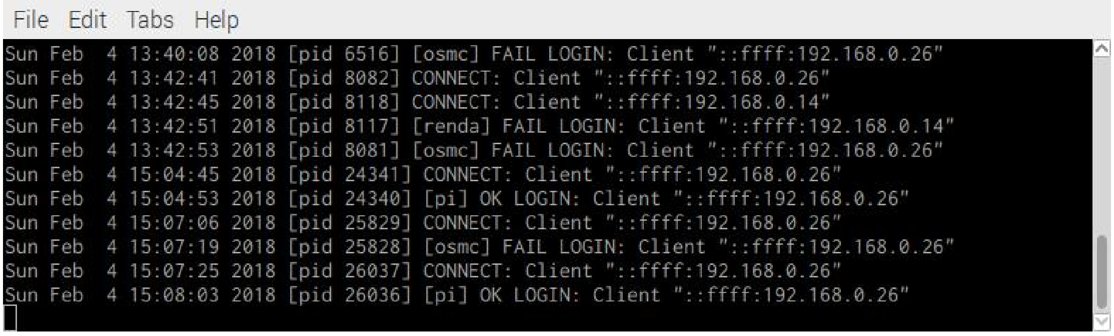
From the ftp log we can find that client 192.168.0.14 has successfully login at 12:46:46



```
** (yad:6567): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Erro
r.ServiceUnknown: The name org.a11y.Bus was not provided by any .service files
Sun Feb 4 12:46:08 2018 [pid 6802] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
./monitor.sh: line 304: notify-send: command not found
./monitor.sh: line 159: notify-send: command not found

###### BlackList #####
######################
#  192.168.0.14 1 #
#####################
```

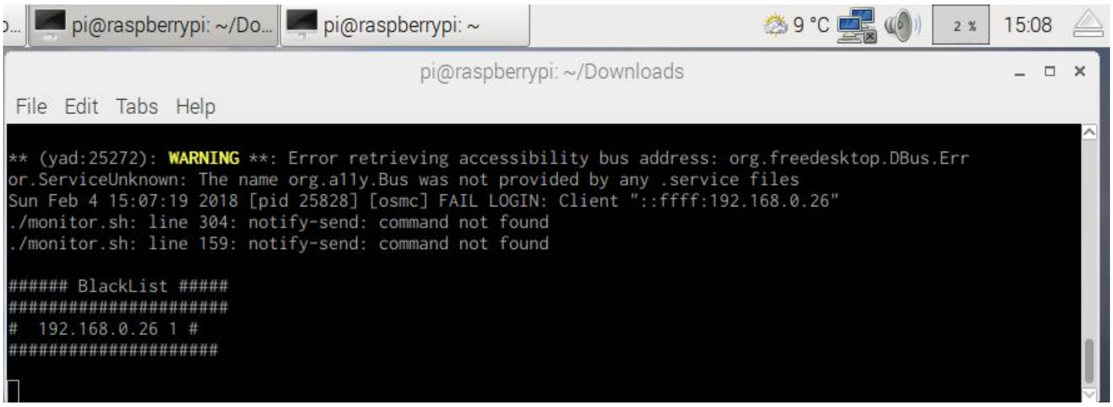
While from the program output message, there is nothing new after 12:46:08

Try again:



```
Sun Feb  4 13:40:08 2018 [pid 6516] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
Sun Feb  4 13:42:41 2018 [pid 8082] CONNECT: Client "::ffff:192.168.0.26"
Sun Feb  4 13:42:45 2018 [pid 8118] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:42:51 2018 [pid 8117] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 13:42:53 2018 [pid 8081] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
Sun Feb  4 15:04:45 2018 [pid 24341] CONNECT: Client "::ffff:192.168.0.26"
Sun Feb  4 15:04:53 2018 [pid 24340] [pi] OK LOGIN: Client "::ffff:192.168.0.26"
Sun Feb  4 15:07:06 2018 [pid 25829] CONNECT: Client "::ffff:192.168.0.26"
Sun Feb  4 15:07:19 2018 [pid 25828] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
Sun Feb  4 15:07:25 2018 [pid 26037] CONNECT: Client "::ffff:192.168.0.26"
Sun Feb  4 15:08:03 2018 [pid 26036] [pi] OK LOGIN: Client "::ffff:192.168.0.26"
```

we can see that at 15:08 there is a time, client successfully logged in ftp server:



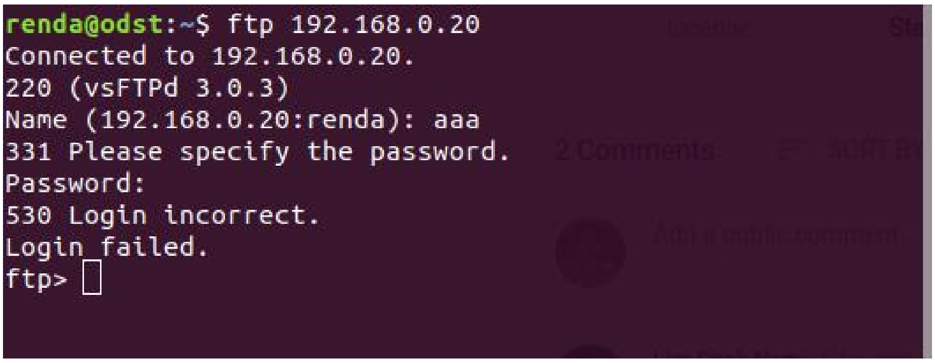
```
** (yad:25272): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Err
or.ServiceUnknown: The name org.a11y.Bus was not provided by any .service files
Sun Feb 4 15:07:19 2018 [pid 25828] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
./monitor.sh: line 304: notify-send: command not found
./monitor.sh: line 159: notify-send: command not found

###### BlackList #####
######################
#  192.168.0.26 1 #
#####################
```

Test pass.

1.2.13 Test 13

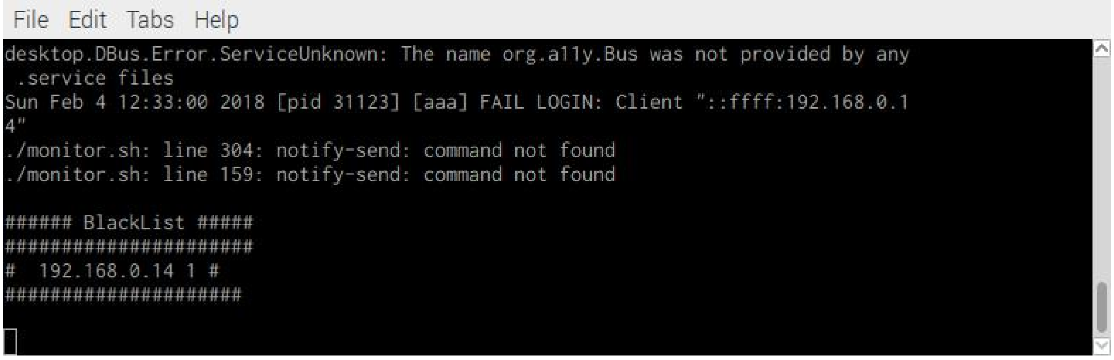
This test case test if one visitor fail to input the correct password for the first time then how the monitor program react.

Run ftp client to log in the server and fail to input password:



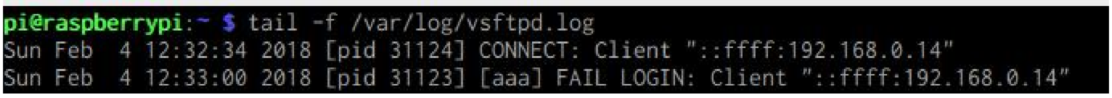
```
renda@odst:~$ ftp 192.168.0.20
Connected to 192.168.0.20.
220 (vsFTPd 3.0.3)
Name (192.168.0.20:renda): aaa
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp> 
```

The server side monitor would output the current blacklist:



```
desktop.DBus.Error.ServiceUnknown: The name org.a11y.Bus was not provided by any
 .service files
Sun Feb 4 12:33:00 2018 [pid 31123] [aaa] FAIL LOGIN: Client "::ffff:192.168.0.1
4"
./monitor.sh: line 304: notify-send: command not found
./monitor.sh: line 159: notify-send: command not found

###### BlackList #####
######################
#  192.168.0.14 1 #
#####################
```

The ftp log would be captured



```
pi@raspberrypi:~ $ tail -f /var/log/vsftpd.log
Sun Feb  4 12:32:34 2018 [pid 31124] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:33:00 2018 [pid 31123] [aaa] FAIL LOGIN: Client "::ffff:192.168.0.14"
```

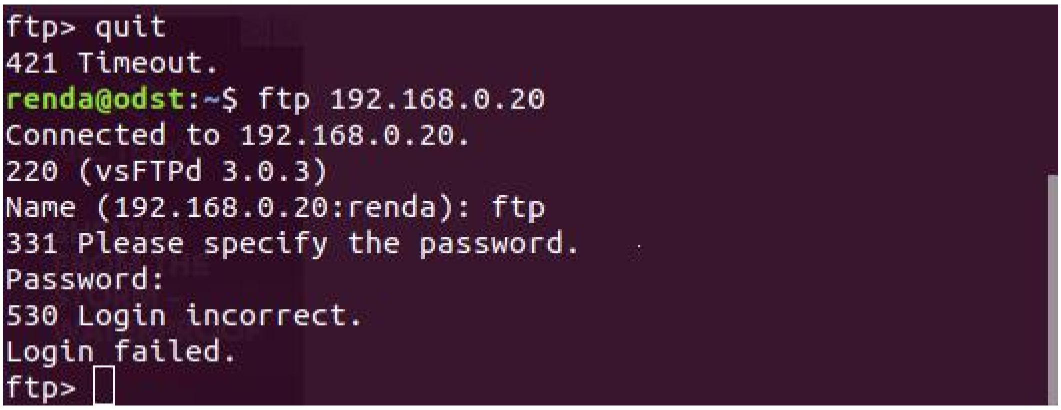
We can find the record by checking time 12:33:00

Test passed.

1.2.14 Test 14

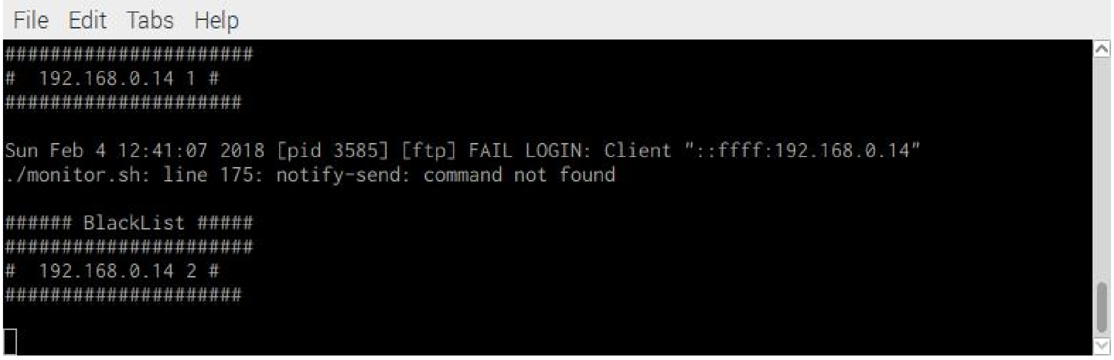
This test case test if one visitor fail to input the correct password for the second time then how the monitor program react.

Run ftp client to log in the server and fail to input password again:



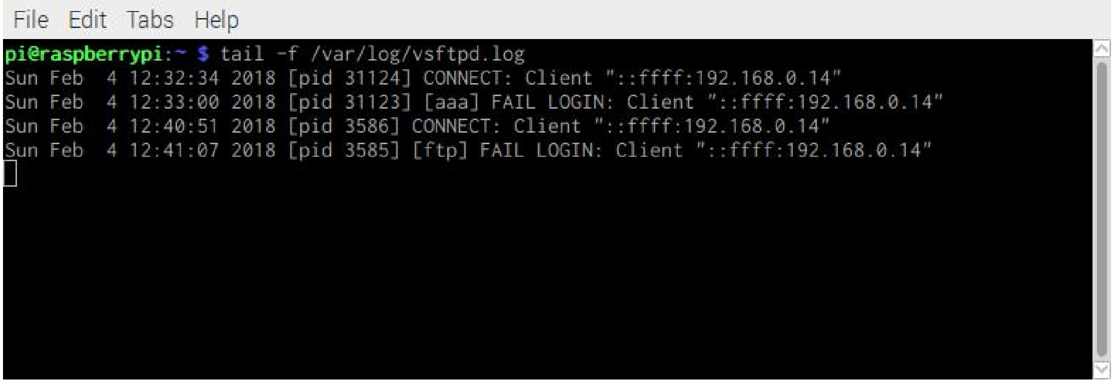
```
ftp> quit
421 Timeout.
renda@odst:~$ ftp 192.168.0.20
Connected to 192.168.0.20.
220 (vsFTPd 3.0.3)
Name (192.168.0.20:renda): ftp
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp> 
```

The server side monitor would renew the current blacklist:



```
######################
#  192.168.0.14 1 #
#####################

Sun Feb 4 12:41:07 2018 [pid 3585] [ftp] FAIL LOGIN: Client "::ffff:192.168.0.14"
./monitor.sh: line 175: notify-send: command not found

###### BlackList #####
######################
#  192.168.0.14 2 #
#####################

```

The ftp log would be captured



```
pi@raspberrypi:~ $ tail -f /var/log/vsftpd.log
Sun Feb  4 12:32:34 2018 [pid 31124] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:33:00 2018 [pid 31123] [aaa] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 12:40:51 2018 [pid 3586] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:41:07 2018 [pid 3585] [ftp] FAIL LOGIN: Client "::ffff:192.168.0.14"
```

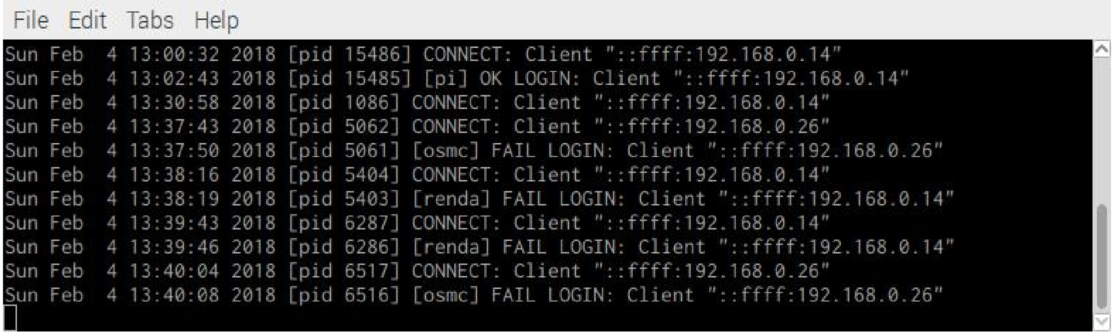
We can find the record by checking time 12:41:07

Test passed.

1.2.15 Test 15

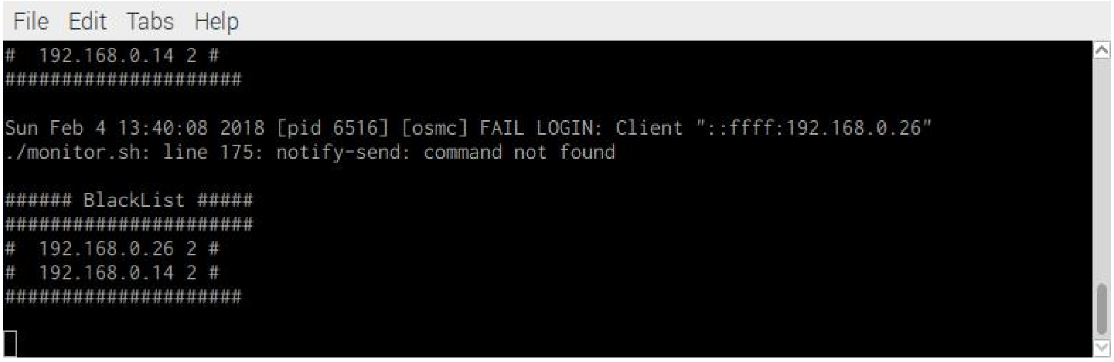
This test case test if more than one visitor fail to input the correct password what the monitor would respond for it.

After experience multiple visitor input wrong password the ftp log shows:



```
Sun Feb  4 13:00:32 2018 [pid 15486] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:02:43 2018 [pid 15485] [pi] OK LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 13:30:58 2018 [pid 1086] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:37:43 2018 [pid 5062] CONNECT: Client "::ffff:192.168.0.26"
Sun Feb  4 13:37:50 2018 [pid 5061] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
Sun Feb  4 13:38:16 2018 [pid 5404] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:38:19 2018 [pid 5403] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 13:39:43 2018 [pid 6287] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:39:46 2018 [pid 6286] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 13:40:04 2018 [pid 6517] CONNECT: Client "::ffff:192.168.0.26"
Sun Feb  4 13:40:08 2018 [pid 6516] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
```

And the monitor would output



```
#  192.168.0.14 2 #
#####################

Sun Feb 4 13:40:08 2018 [pid 6516] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
./monitor.sh: line 175: notify-send: command not found

###### BlackList #####
######################
#  192.168.0.26 2 #
#  192.168.0.14 2 #
#####################
```

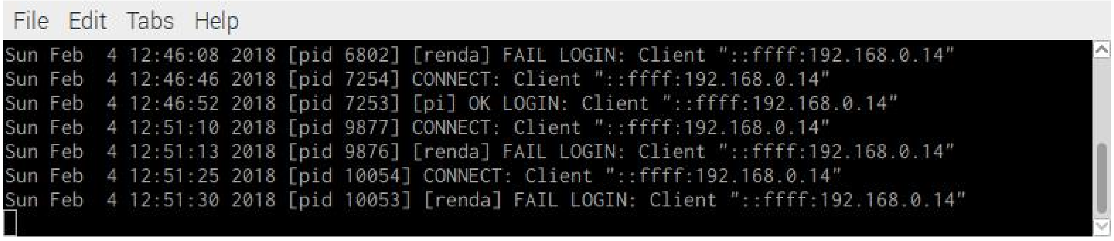
We can find that there are two ip in the current blacklist, and both of them try the password twice.

Test passed.

1.2.16 Test 16

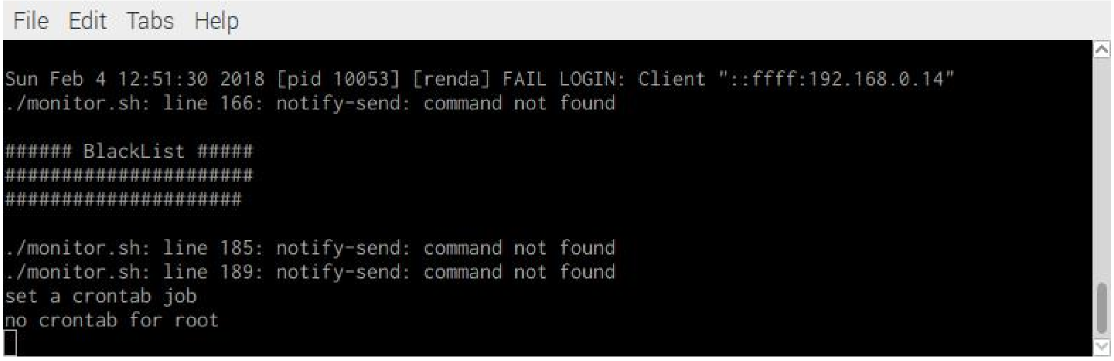
This test case test if one visitor try many times that over the threshold then what the monitor would respond for it.

After trying for many times at least get the threshold attempt times by one, the vsftp log:



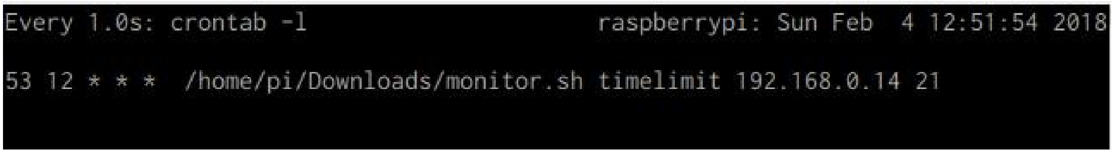
```
Sun Feb  4 12:46:08 2018 [pid 6802] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 12:46:46 2018 [pid 7254] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:46:52 2018 [pid 7253] [pi] OK LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 12:51:10 2018 [pid 9877] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:51:13 2018 [pid 9876] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 12:51:25 2018 [pid 10054] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:51:30 2018 [pid 10053] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
```

Monitor output

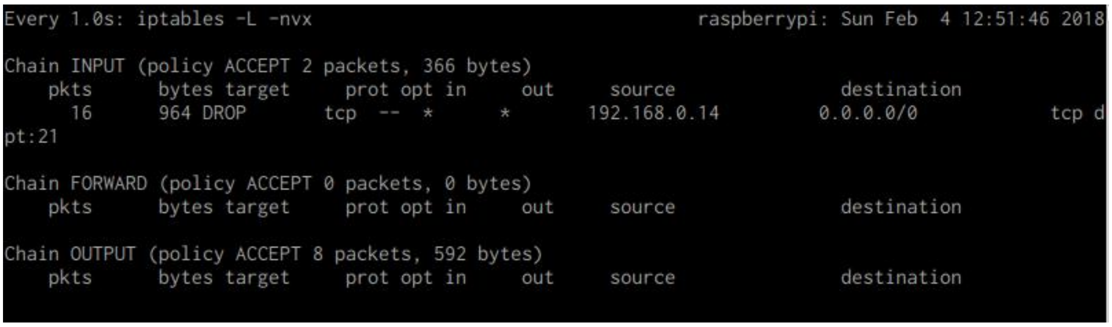


```
Sun Feb 4 12:51:30 2018 [pid 10053] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
./monitor.sh: line 166: notify-send: command not found

###### BlackList #####
######################
#####################

./monitor.sh: line 185: notify-send: command not found
./monitor.sh: line 189: notify-send: command not found
set a crontab job
no crontab for root
```

We can find that the blacklist is cleared;
at the same time check crontab and iptables rule:



```
Every 1.0s: crontab -l                      raspberrypi: Sun Feb  4 12:51:54 2018

53 12 * * *  /home/pi/Downloads/monitor.sh timelimit 192.168.0.14 21
```



```
Every 1.0s: iptables -L -nvx                                     raspberrypi: Sun Feb  4 12:51:46 2018

Chain INPUT (policy ACCEPT 2 packets, 366 bytes)
    pkts      bytes target     prot opt in     out     source               destination
      16      964 DROP       tcp  --  *      *       192.168.0.14         0.0.0.0/0            tcp d
pt:21

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 8 packets, 592 bytes)
    pkts      bytes target     prot opt in     out     source               destination
```

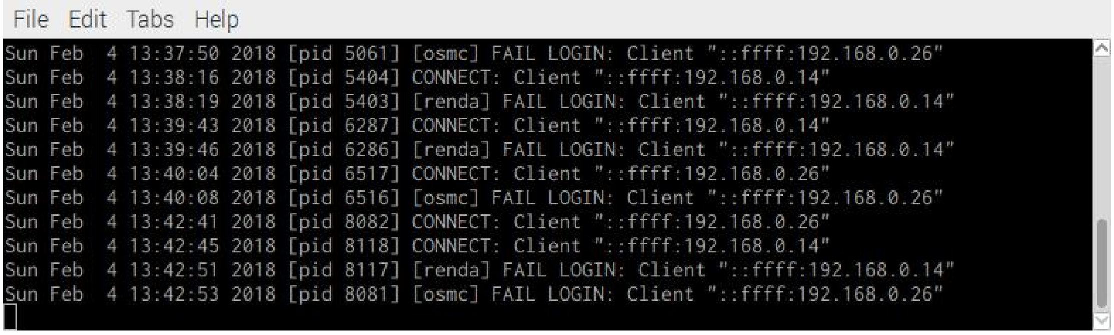
So we can see that the monitor add a firewall blocking rule and a crontab task in this process.

Test passed.

1.2.17 Test 17

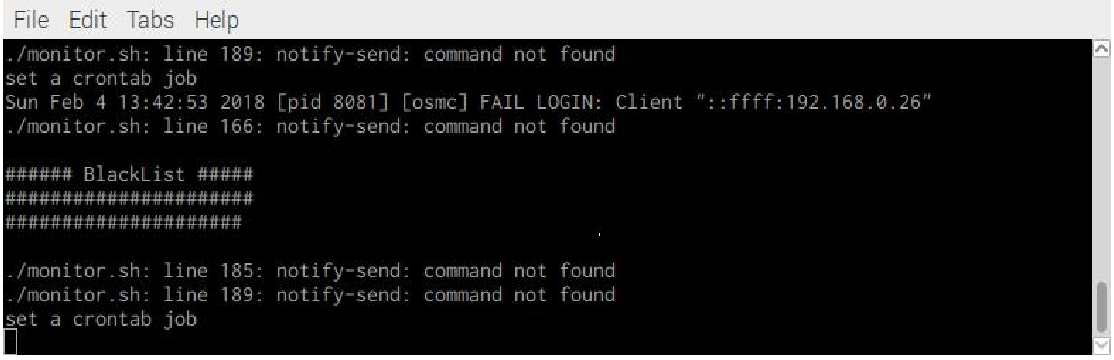
This test case test if more than one visitors try many times that over the threshold then what the monitor would respond for it.

After trying for many times at least get the threshold attempt times by multiple visitors, the vsftp log:



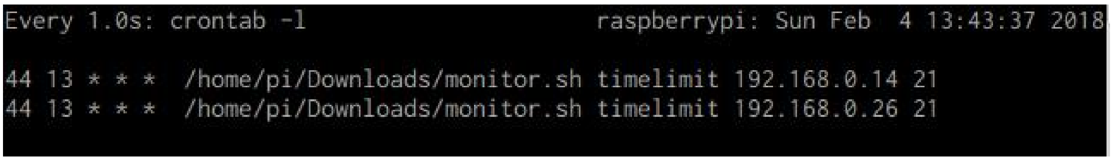
```
Sun Feb  4 13:37:50 2018 [pid 5061] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
Sun Feb  4 13:38:16 2018 [pid 5404] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:38:19 2018 [pid 5403] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 13:39:43 2018 [pid 6287] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:39:46 2018 [pid 6286] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 13:40:04 2018 [pid 6517] CONNECT: Client "::ffff:192.168.0.26"
Sun Feb  4 13:40:08 2018 [pid 6516] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
Sun Feb  4 13:42:41 2018 [pid 8082] CONNECT: Client "::ffff:192.168.0.26"
Sun Feb  4 13:42:45 2018 [pid 8118] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:42:51 2018 [pid 8117] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 13:42:53 2018 [pid 8081] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
```

Monitor output

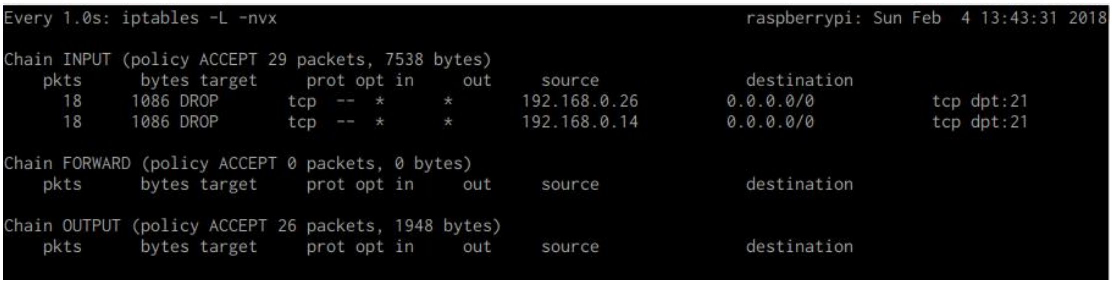


```
./monitor.sh: line 189: notify-send: command not found
set a crontab job
Sun Feb 4 13:42:53 2018 [pid 8081] [osmc] FAIL LOGIN: Client "::ffff:192.168.0.26"
./monitor.sh: line 166: notify-send: command not found

###### BlackList #####
######################
#####################

./monitor.sh: line 185: notify-send: command not found
./monitor.sh: line 189: notify-send: command not found
set a crontab job
```

We can find that the blacklist is cleared;
at the same time check crontab and iptables rule:



```
Every 1.0s: crontab -l                    raspberrypi: Sun Feb  4 13:43:37 2018

44 13 * * *  /home/pi/Downloads/monitor.sh timelimit 192.168.0.14 21
44 13 * * *  /home/pi/Downloads/monitor.sh timelimit 192.168.0.26 21
```



```
Every 1.0s: iptables -L -nvx                                       raspberrypi: Sun Feb  4 13:43:31 2018

Chain INPUT (policy ACCEPT 29 packets, 7538 bytes)
    pkts      bytes target     prot opt in     out     source               destination
      18      1086 DROP       tcp  --  *      *       192.168.0.26         0.0.0.0/0            tcp dpt:21
      18      1086 DROP       tcp  --  *      *       192.168.0.14         0.0.0.0/0            tcp dpt:21

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 26 packets, 1948 bytes)
    pkts      bytes target     prot opt in     out     source               destination
```

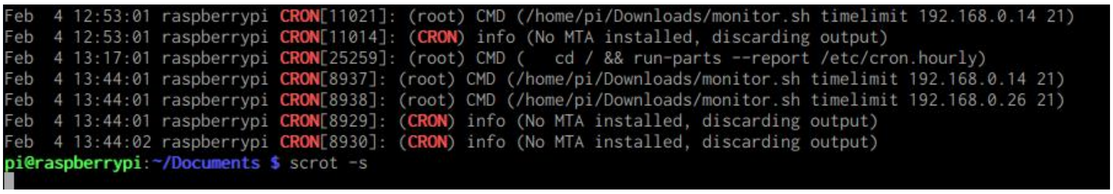
So we can see that the monitor add a firewall blocking rule and a crontab task in this process for both ip.

Test passed.

1.2.18 Test 18

This test case test how crontab task activate the application after monitor insert a rule into crontab.

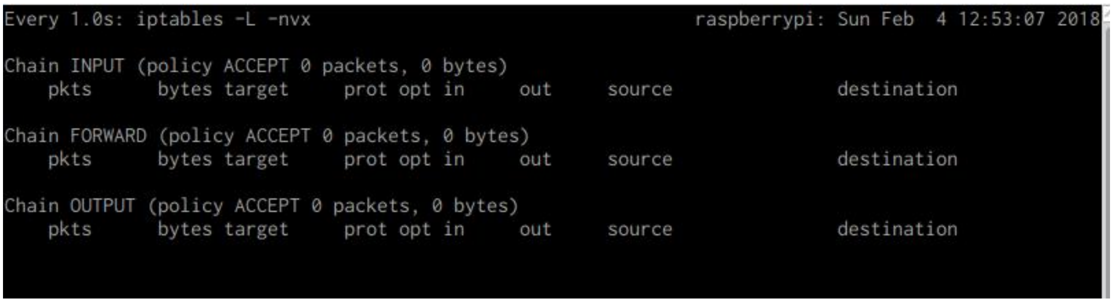
When the task start time pass, the crontab log record:



```
Feb  4 12:53:01 raspberrypi CRON[11021]: (root) CMD (/home/pi/Downloads/monitor.sh timelimit 192.168.0.14 21)
Feb  4 12:53:01 raspberrypi CRON[11014]: (CRON) info (No MTA installed, discarding output)
Feb  4 13:17:01 raspberrypi CRON[25259]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Feb  4 13:44:01 raspberrypi CRON[8937]: (root) CMD (/home/pi/Downloads/monitor.sh timelimit 192.168.0.14 21)
Feb  4 13:44:01 raspberrypi CRON[8938]: (root) CMD (/home/pi/Downloads/monitor.sh timelimit 192.168.0.26 21)
Feb  4 13:44:01 raspberrypi CRON[8929]: (CRON) info (No MTA installed, discarding output)
Feb  4 13:44:02 raspberrypi CRON[8930]: (CRON) info (No MTA installed, discarding output)
pi@raspberrypi:~/Documents $ scrot -s
```

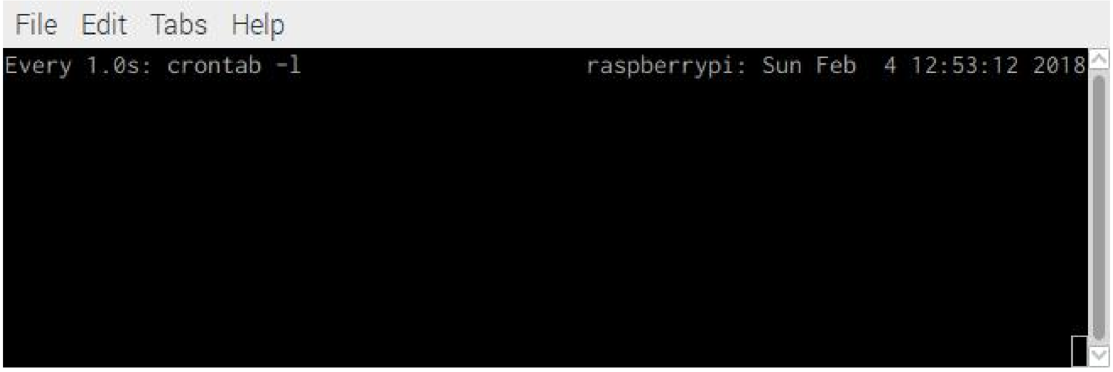
This cron task would clear the firewall rule:

After running the task, this task would be clear from the crontab:



```
Every 1.0s: iptables -L -nvx                              raspberrypi: Sun Feb  4 12:53:07 2018

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
```

And then clear itself from crontab:



```
Every 1.0s: crontab -l                              raspberrypi: Sun Feb  4 12:53:12 2018
```

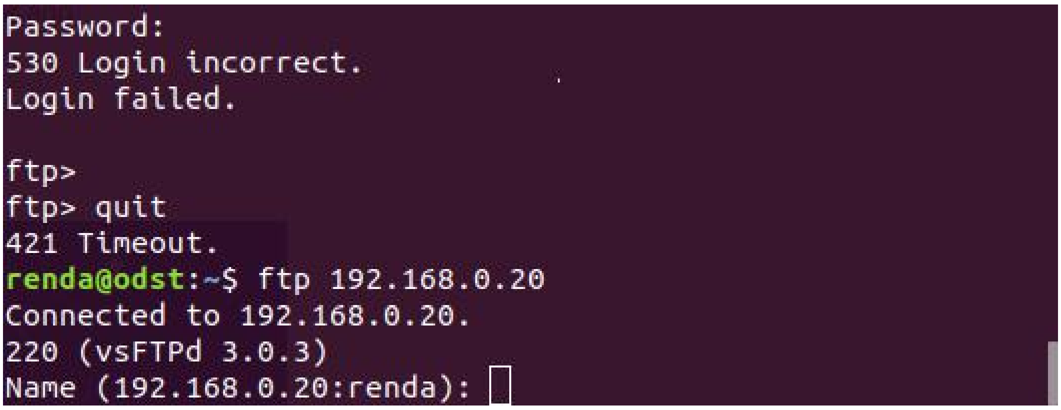
So crontab has worked.

Test passed.

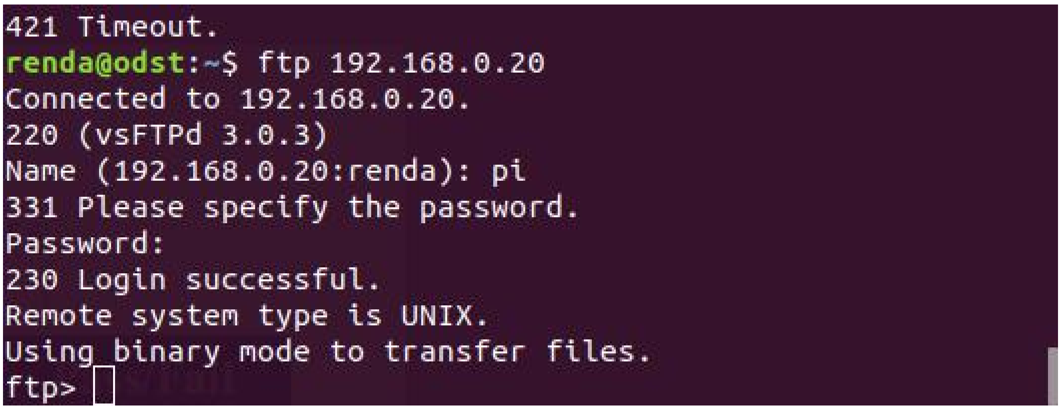
1.2.19 Test 19

This test case test if after blocking period the visitor can access FTP service again

After the blocking time, visitor try to log in the ftp again:

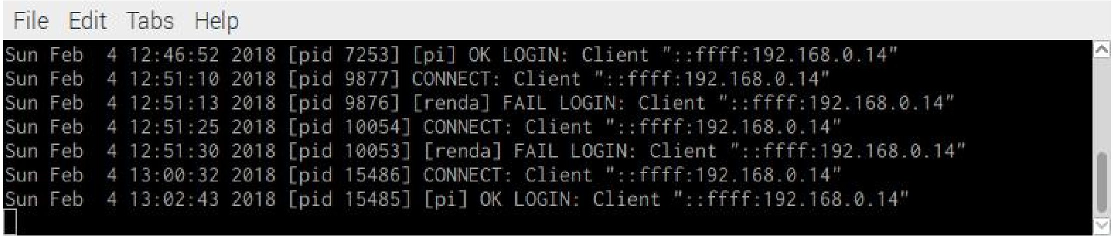


```
Password:
530 Login incorrect.
Login failed.

ftp>
ftp> quit
421 Timeout.
renda@odst:~$ ftp 192.168.0.20
Connected to 192.168.0.20.
220 (vsFTPd 3.0.3)
Name (192.168.0.20:renda):
```



```
421 Timeout.
renda@odst:~$ ftp 192.168.0.20
Connected to 192.168.0.20.
220 (vsFTPd 3.0.3)
Name (192.168.0.20:renda): pi
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

And the vsftp log shows:



```
Sun Feb  4 12:46:52 2018 [pid 7253] [pi] OK LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 12:51:10 2018 [pid 9877] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:51:13 2018 [pid 9876] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 12:51:25 2018 [pid 10054] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 12:51:30 2018 [pid 10053] [renda] FAIL LOGIN: Client "::ffff:192.168.0.14"
Sun Feb  4 13:00:32 2018 [pid 15486] CONNECT: Client "::ffff:192.168.0.14"
Sun Feb  4 13:02:43 2018 [pid 15485] [pi] OK LOGIN: Client "::ffff:192.168.0.14"
```

Test passed.