8006 Assignment 3

Design & Documentation

Eric Wu, A00961904
Hong Kit, Wu A00968591

BTech 2020 Winter

Table of contents

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.

Approach

We will use shell script for the implementation. We will use **Netfilter** for generating IP blocking activity. There will be a watch script (**watch-script.sh**) monitoring the log file (e.g. /var/log/secure), and a stop script (**stop-script.sh**) that can be used later to terminate the watch script running in background. If there's a new entry in the log file, the watch script will grab it, process the new entry to determine if it is a password fail attempt. The result will be stored in a database to be used later when determining an IP should be blocked or unblock.

Application design

Pseudo implementation

(**watch-script.sh**)

```
while (TRUE) {
  wait for new entry in log file and store in variable line;
  if the variable line consists of the key phrase "Failed password" {
    grep the IP from the entry and store it in variabel IP;
  }
  if the variable IP is not empty {
    if the ip exists in the database {
      increment the column value FAILED_ATTEMPT of the existing entry.
    } else {
      add a new entry into database with value IP for column IP and 1 for column
      FAILED_ATTEMPT.
    }
    grab an entry from database that has IP as primary key, and grab the column value;
    FAILLED_ATTEMPT and store it in the variable COUNT.
    if COUNT greater or equal to maximum number of fail allowed {
      delete the entry from database;
      block any incoming traffic from that ip;
      schedule a job to run after user specify duration passed, the job will unblock the IP;
    }
  }
}
```

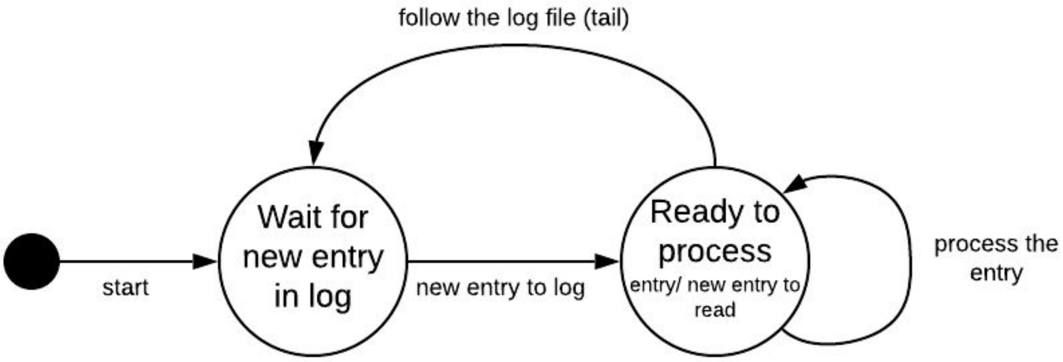
(**watch-script.sh**)

Kill the running process generate by watch script

(**setup-script**)

Create a database for storing IP and failed attempt, column value are IP, FAILED_ATTEMPT
Create a user define chain for keeping block rules

State diagram



How to run

The application consist of four files

- watch-script.sh
- stop-script.sh
- assign3.conf
- setup.sh

Before starting the application, fill in the variables in assign3.conf as follow

- MAX_FAILED_ATTEMPT (maximum number of fail attempt before blocking that ip)
- BLOCK_DURATION_MIN (blocking duration)
- LOG_FILE_LOCATION (location of the file to monitor, etc. /var/log/secure)

Run the setup script once to create database and user defined chain in iptables

```
./setup.sh
```

After, run the watch script, which will run in the background.

```
./stop-script.sh
```

Run the stop script to stop the application completely

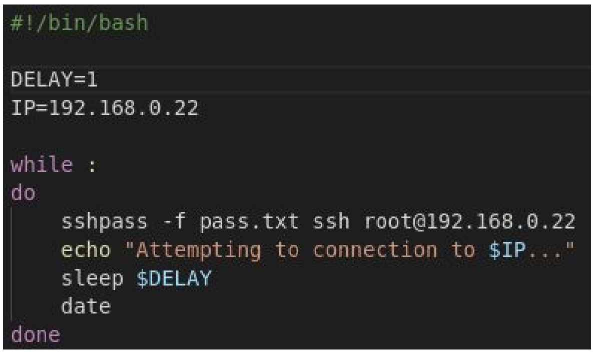
```
./watch-script.sh
```

Testing

Test environment setup

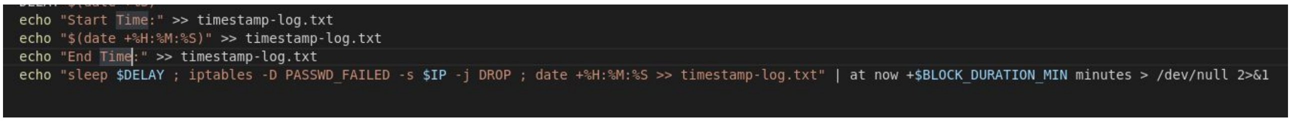
The testing environment consists of two machines, one machine run as a remote login host, another run as a remote login client.

Sample test script on client



```
#!/bin/bash

DELAY=1
IP=192.168.0.22

while :
do
    sshpass -f pass.txt ssh root@192.168.0.22
    echo "Attempting to connection to $IP..."
    sleep $DELAY
    date
done
```

Script for recording timestamp on remove login server



```
echo "Start Time:" >> timestamp-log.txt
echo "$(date +%H:%M:%S)" >> timestamp-log.txt
echo "End Time:" >> timestamp-log.txt
echo "sleep $DELAY ; iptables -D PASSWD_FAILED -s $IP -j DROP ; date +%H:%M:%S >> timestamp-log.txt" | at now +$BLOCK_DURATION_MIN minutes > /dev/null 2>&1
```

Test cases & results

Remote login client IP: 192.169.0.21
Remote login server IP: 192.169.0.22

Test Case #	Description	Tool	Expected Result	Result Pass/ Failed
Case 1	When block threshold is set to 1, and block duration is set to 1. Verify that the IP gets blocked for 1 min after the remote login client machine fails to password login into the remote login server 1 time.	ssh, iptables	● A drop rule should be added on the iptables for that IP ● The drop rule should be deleted after 1 min.	**Passed**
Case 2	When block threshold is set to 10, and block duration is set to 10. Verify that the IP gets blocked for 10 mins after the remote login client machine fails to password login into the remote login server 10 times.	ssh, iptables	● A drop rule should be added on the iptables for that IP ● The drop rule should be deleted after 10 mins.	**Passed**

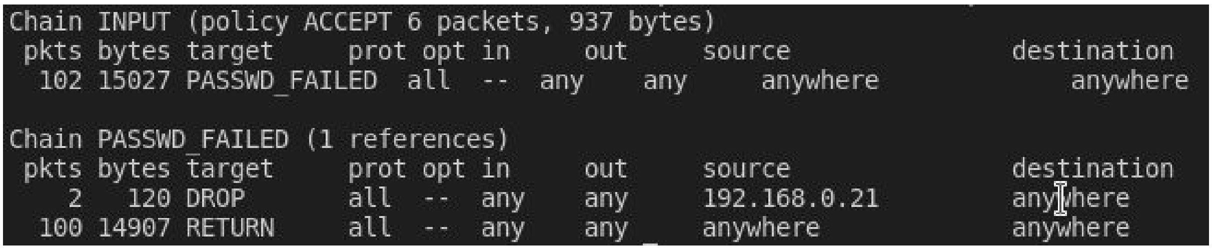
Case 3	When block threshold is set to 2, and block duration is set to 2. The remote login client machine attempts to ssh into the remote login server every 10 mins for 2 times . Verify that the IP gets blocked for 2 mins after the remote login client machine fails to password login into the remote login server 2 times. (slow scan)	ssh, iptables	● A drop rule should be added on the iptables for that IP ● The drop rule should be deleted after 2 mins.	**Passed**
Case 4	When the block threshold is set to 3, and the block duration is not set. Verify that the IP gets blocked permanently after the remote login client machine fails to password login into the remote login server 3 times.	ssh, iptables	● A drop rule will be added on the iptables for that IP permanently. .	**Passed**

Supporting evidence

Case 1:

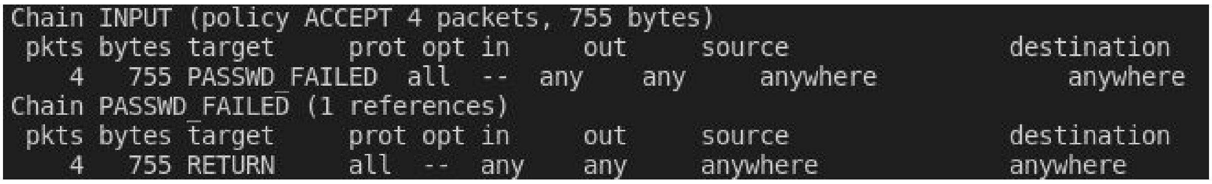
Remote login client IP: 192.169.0.21
Remote login server IP: 192.169.0.22

iptables listing on remove login host, the **DROP** rule in **PASSWD_FAILED** chain shows that the remote login host (192.168.0.21) has been blocked.



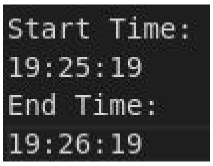
```
Chain INPUT (policy ACCEPT 6 packets, 937 bytes)
 pkts bytes target     prot opt in     out     source               destination
  102 15027 PASSWD_FAILED  all  --  any    any     anywhere             anywhere

Chain PASSWD_FAILED (1 references)
 pkts bytes target     prot opt in     out     source               destination
    2   120 DROP       all  --  any    any     192.168.0.21         anywhere
  100 14907 RETURN     all  --  any    any     anywhere             anywhere
```

After 1 min, the **DROP** rule for 192.168.0.21 was removed from the PASSWD_FAILED chain



```
Chain INPUT (policy ACCEPT 4 packets, 755 bytes)
 pkts bytes target     prot opt in     out     source               destination
    4   755 PASSWD_FAILED  all  --  any    any     anywhere             anywhere
Chain PASSWD_FAILED (1 references)
 pkts bytes target     prot opt in     out     source               destination
    4   755 RETURN     all  --  any    any     anywhere             anywhere
```

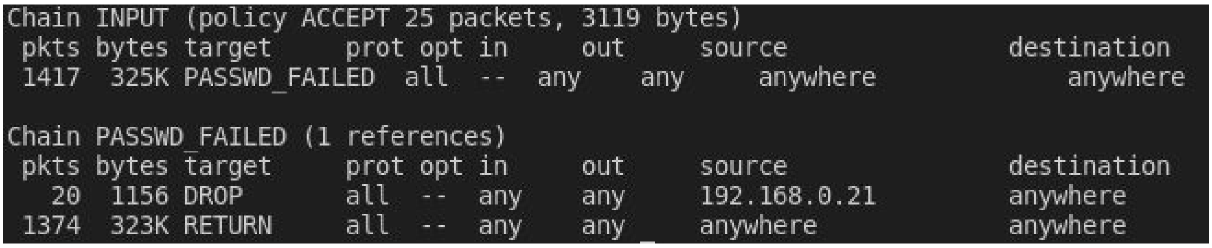
time stamp taken on remove login host, the start time indicates start time of blocking, the end time indicates unblock time (block duration of 1 minutes).



```
Start Time:
19:25:19
End Time:
19:26:19
```

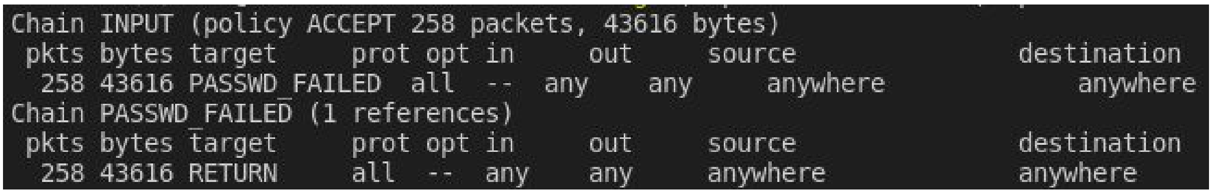
Case 2:
Remote login client IP: 192.169.0.21
Remote login server IP: 192.169.0.22

iptables listing on remove login host, the **DROP** rule in **PASSWD_FAILED** chain shows that the remote login host (192.168.0.21) has been blocked.



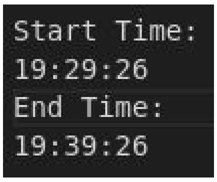
```
Chain INPUT (policy ACCEPT 25 packets, 3119 bytes)
 pkts bytes target     prot opt in     out     source               destination
 1417  325K PASSWD_FAILED  all  --  any    any     anywhere             anywhere

Chain PASSWD_FAILED (1 references)
 pkts bytes target     prot opt in     out     source               destination
   20  1156 DROP       all  --  any    any     192.168.0.21         anywhere
 1374  323K RETURN     all  --  any    any     anywhere             anywhere
```

After 10 min, the **DROP** rule for 192.168.0.21 was removed from the **PASSWD_FAILED** chain



```
Chain INPUT (policy ACCEPT 258 packets, 43616 bytes)
 pkts bytes target     prot opt in     out     source               destination
  258 43616 PASSWD_FAILED  all  --  any    any     anywhere             anywhere
Chain PASSWD_FAILED (1 references)
 pkts bytes target     prot opt in     out     source               destination
  258 43616 RETURN     all  --  any    any     anywhere             anywhere
```

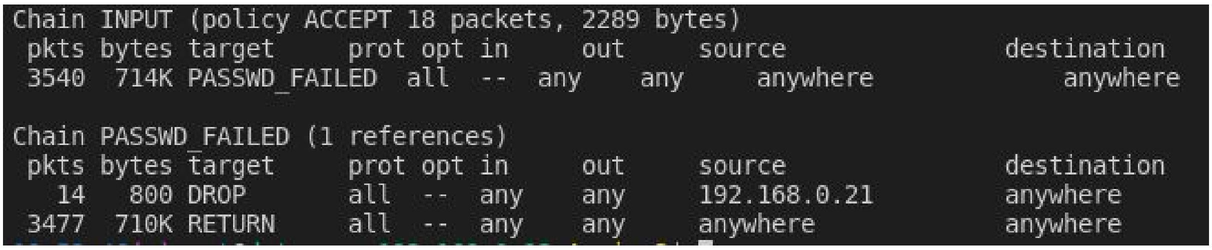
time stamp for on remove login host, the start time indicates start time of blocking, the end time indicates unblock time (block duration of 10 minutes).



```
Start Time:
19:29:26
End Time:
19:39:26
```

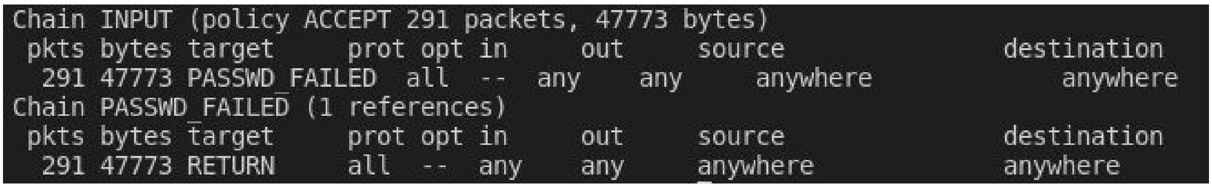
Case 3:
Remote login client IP: 192.169.0.21
Remote login server IP: 192.169.0.22

iptables listing on remove login host, the **DROP** rule in **PASSWD_FAILED** chain shows that the remote login host (192.168.0.21) has been blocked.



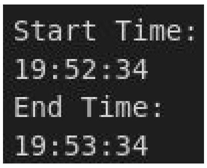
```
Chain INPUT (policy ACCEPT 18 packets, 2289 bytes)
 pkts bytes target     prot opt in     out     source               destination
 3540  714K PASSWD_FAILED  all  --  any    any     anywhere             anywhere

Chain PASSWD_FAILED (1 references)
 pkts bytes target     prot opt in     out     source               destination
   14   800 DROP       all  --  any    any     192.168.0.21         anywhere
 3477  710K RETURN     all  --  any    any     anywhere             anywhere
```

After 1 min, the **DROP** rule for 192.168.0.21 was removed from the **PASSWD_FAILED** chain



```
Chain INPUT (policy ACCEPT 291 packets, 47773 bytes)
 pkts bytes target     prot opt in     out     source               destination
  291 47773 PASSWD_FAILED  all  --  any    any     anywhere             anywhere
Chain PASSWD_FAILED (1 references)
 pkts bytes target     prot opt in     out     source               destination
  291 47773 RETURN     all  --  any    any     anywhere             anywhere
```

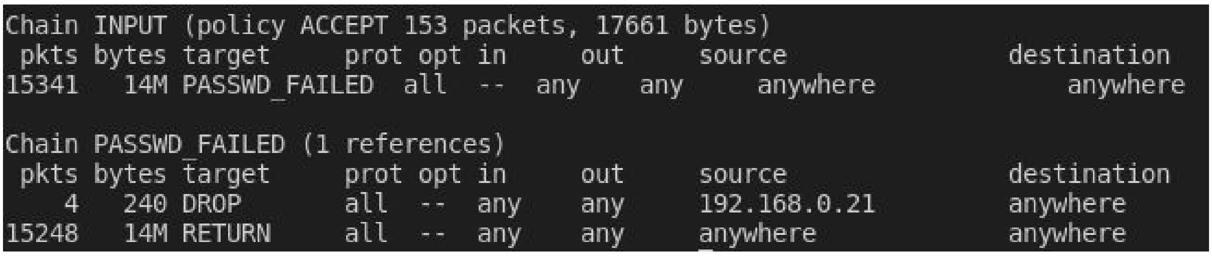
time stamp for on remove login host, the start time indicates start time of blocking, the end time indicates unblock time (block duration of 1 minutes).



```
Start Time:
19:52:34
End Time:
19:53:34
```

Case 4:
Remote login client IP: 192.169.0.21
Remote login server IP: 192.169.0.22

iptables listing on remove login host, the **DROP** rule in **PASSWD_FAILED** chain shows that the remote login host (192.168.0.21) has been blocked permanently.



```
Chain INPUT (policy ACCEPT 153 packets, 17661 bytes)
 pkts bytes target     prot opt in     out     source               destination
15341   14M PASSWD_FAILED  all  --  any    any     anywhere             anywhere

Chain PASSWD_FAILED (1 references)
 pkts bytes target     prot opt in     out     source               destination
    4   240 DROP       all  --  any    any     192.168.0.21         anywhere
15248   14M RETURN     all  --  any    any     anywhere             anywhere
```