miniban - a lightweight version of fail2ban

fail2ban is a software package written in Python and released under the GNU Public License 2.0 that bans an IP address after repeated failed login attempts. It normally runs as a daemon process. The objective of this part of the project is to write a simple verison of fail2ban, called **miniban**, using the Bash scripting language.

There are three parts to miniban:

1. **miniban.sh** Main script - watch for system authorisation events relating to SSH (secure shell), keep track of the number of failures per IP address, and when this is greater than or equal to 3, ban the IP address (see next point)
2. **ban.sh** Ban an IP address using iptables and add the IP address together with a ban timestamp to a persistent flat database file miniban.db (see format below)
3. **unban.sh** Periodically check banned IPs in miniban.db and remove them if the ban has expired (10 minutes). Remove the iptables rule for the IP address

Write a separate Bash script for each point above. Separating functionality into three scripts allows an administrator to ban or unban IP addresses manually.

Banfile format

The format of the banfile miniban.db should be *<IP>,<TIMESTAMP>*:

10.0.2.15,1572887890
127.0.0.1,1572887940

The TIMESTAMP field is the number of seconds since Jan 1 1970. See the date manpage for more information about date formats.

Event logs

Depending on the Linux distribution, authorisation events can be accessed several places. The most common is via the journalctl command. An alternative on many systems is reading the logfile /var/log/auth.log directly. For the former method, to filter for only SSH log messages:

$ journalctl -u ssh

See the journalctl manpage for more information.

Tips

- Read about iptables, specifically how to add and remove rules from the INPUT chain to block or reject traffic matching a specified IP address
- Install openssh-server (sudo apt install openssh-server) and ensure that it is running by logging in locally:
  - ssh localhost
- Use the flock command to synchronise access to the ban file miniban.db (parts 2 and 3 above will be running asynchronously)
- Use the tail command to monitor activity in the logfile /var/log/auth.log
- Use the Bash builtin command "read" to read lines from files:
  - IFS="," read IP TIMESTAMP < miniban.db; echo $IP $TIMESTAMP

    - Note IFS above, internal field separator set to be a comma, read about it in the Bash manpage
- Use regular expressions to extract the IP address from the SSH event log
- Store failed attempts per IP address in a Bash associative array where the index is the IP address and the value is the number of failed login attempts
- Use the date command to deal with timestamps: date +%s
- Use Bash syntax $((10 - 5)) to do simple arithmetic: echo $((10 - 5))
- Use Bash command substitution where necesssary, it can simplify your script
- Run the script as the root user, so that you have permissions to run iptables and view SSH event logs

Bonus credit

Implement the following for bonus credit:

1. IP address whitelist - addresses stored in a file miniban.whitelist, one IP address per line, are never banned (e.g., 127.0.0.1)
2. Clean-up all child processes on program termination - When CTRL-C is pressed, any child processes should also be terminated (hint: see the Bash builtin trap command)