SSH Intrusion Detector Design

Gary Khodayari
February 27sth, 2022

Application Process Steps

g8keep3r.sh

g8keep3r.sh is my entry script and the only script the user will directly use to use the program. The idea is for the script to prompt the user if the wrong arguments are used to assist them with using the program.

I will two main functions done with g8keep3r script, remove and add user. When adding a user, two extra arguments for the duration and failed attempt limit are required. Add user will add a cronjob to the root crontab that pass the username, duration and attempt limit as arguments to the parser script. And remove user will remove the cronjob if it exists.

To be able to modify the root crontab we will need this script to be always ran as root.

In addition, g8keep4r script will also setup the iptables to read ipset groups.

parser.sh

The purpose of parser.sh is to get information with a timestamp from users who enter the wrong password.

The initial idea was for the parser to act as a script that runs every minute through the crontab application though first parse the /var/log/auth.log file and only grab the failed attempt logs relating to the both the user and ssh connections and put them in the
<line format : Feb 6 02:52:49 10.0.0.228>
so it can easily be read after. I will also make the program to clear the log file if it finds a successful attempt. This allows us to add a pseudo “if saucerful ignore the failed attempt” mechanism.

After that the parser would save the useful information in a file that can be read by the next part. This allows us to be to read the parser logfile ourselves in case of debugging needs.

The second of the parser is the parts that looks to see if there are malicious terrific trying to connect to our host. The plan is for the parser to first convert the date on each line to the UNIX time format (second from 1970 jan 1st) since its much easier to calculate time difference from. Then it compares each line timestamp to the current UNIX time and if they from less than 60 seconds ago it would save in an array. This basically calculated all the failed attempts done in the last minute. After that we only to read the array to see if there are more than 5 entries for each user and if so, we ban the ip trying to brute force our host and promp the shell. Parser achieves this by adding malicious ips with the user chose timeout duration to the g8keep3r ipset group that g8keep3r script has already created.