**Ubuntu Security Scripts**

**Scripts**

The following scripts were shared over from my centOS server and were tweaked for my ubuntu server see the centOS document to compare the scripts

**Comp 1 and 2**

The following scripts are for the second portion asking to detect changes in a directory I did **/var/log** and I decided to detect changes in that file I did it in two scripts **compOne.sh** and **compTwo.sh** the first portion of the code is relatively simple and consists of a command an simple command as shown in this screenshot

A screen shot of a computer

Description automatically generated

This takes the hash of all files in **/var/log** using an asterisk for a wild card and appends and overwrites to a file in **/home/dlq03/reports/base.txt** (reports directory is made in the security script see more below)

Next we go to the second part in **compTwo.sh** of the script where I do the same in **compOne.sh** but then compare the file made in **compOne.sh** two the file **compTwo.sh** to compare differences in hash it then appends the output of the command to **/home/dlq03/reports/changes.txt** the if statement checks if there is anything in that file as if the contents are the same the diff command will output nothing if there something in the **changes.txt file** it will echo that changes were detected else it will echo that there were no changes detected

**A computer screen shot of a program

Description automatically generated**Scripts in action:

A screenshot of a computer program

Description automatically generated

(Contents of changes.txt)

**Security Script**

The next script is the main security script detecting red flag behavior as seen below

A screenshot of a computer program

Description automatically generated

The first thing this script does is see if the directory **reports** exists this where the files that will be detecting different portions of the code will be stored the if statement checks if the directory exists it will echo a message saying that the script will begin or else it will make the directory and echo a message saying so assuming it is for first time use it will then do the following procedures make sure you have sudo privileges.

**sudo grep -E "0[0-6]:[0-9]{2}:[0-9]{2}" /var/log/auth.log >> "$DIR/12to6.txt/**

This command uses searches for the pattern of characters in the quotes which is a regular expression to find the time between 12 AM to 6 AM in the file **/var/log/auth.log** which is where authentication related activity is stored this is concerning because ideally speaking I should and other users should be using it during the day time so there is no reason for anyone to be using this server during that time anything the command finds will be output to **reports/12to6.txt**

A screen shot of a computer

Description automatically generated

(Contents from 12to6.txt)

**sudo last -f /var/log/btmp | grep -v -e "10.0.0.88" -e "10.0.0.225" >> "$DIR/unknownusers.txt"**

This command uses **last -f** in the file **/var/log/btmp** which is where failed logins are stored I then pipe and use a grep commands to not get outputs from **10.0.0.88** which is the ip address for my centOS server address which I trust and and my **10.0.0.225** address which is my host address. This portion is meant to detect attempts from unknown ip addresses who could be attempting to access for malicious reasons. to test the effectiveness of this script I made another ubuntu server called serverc and have made a login attempt this command will outputs failed login attempts from that servers ip and not the others and append the output to **unkownusers.txt**

A screen shot of a computer

Description automatically generated

(Results from unknownusers.txt)

**sudo grep -E "password check failed" /var/log/auth.log >> "$DIR/failedlogins.txt"**

**sudo grep "authentication failure" /var/log/auth.log >> "$DIR/reportsauthfail.txt"**

These two commands do similar things but differently the first command on **password check failed** checks for any lines lines in **/var/log/auth.log** which tracks any login related activity this command is used to append any wrong password attempt and appends to the text file **failedlogins.txt**

A screen shot of a computer screen

Description automatically generated

(Results from failedlogins.txt)

The next command is also for failed logins but the quotes **“authentication failure”** is for failed logins as after 3 failed login attempts it stops the login and logs in as a failed login after 3 attempts and you have to ssh again in order to attempt to logins. Tracking both of these are important to see if any malicious actors are trying to access the server so keeping tabs on failed passwords and failed authentication could provide insight on any attempt.

A blurry image of a screen

Description automatically generated

(from reportsauthfail.txt)

**sudo grep "closed" /var/log/auth.log >> "$DIR/closed.txt"**

This command uses grep to search for the pattern **closed**  in **/var/log/auth.log** this will mainly show any closed sessions but also in the event of a ddos attack show any connections closed the contents of this command is appended to **closed.txt**

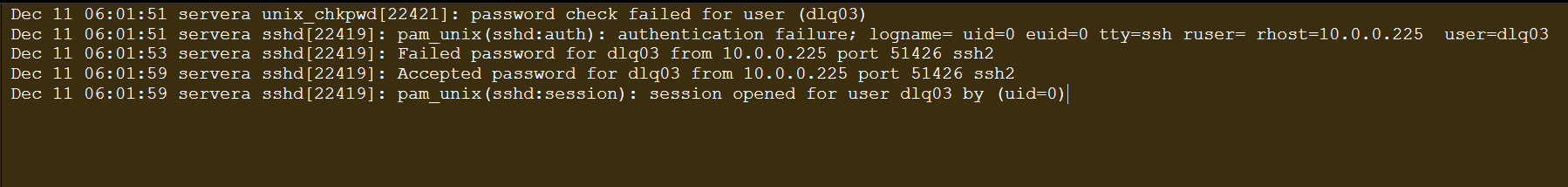
A screen shot of a computer screen

Description automatically generated

(from closed.txt)

**sudo grep "dlara" /var/log/auth.log >> "$DIR/dlara.txt"**

This command appends any login activity for the user dlara into the file **dlara.txt** this account was made for my centos server to connect to but since my centos and ubuntu server do not currently have a reason to connect with each other aside from communication this account is abandoned and not used and as such there should not be much activity coming from such user so it may be worth monitoring activity.



**Conclusion**

Overall while these scripts are serviceable with what they are able to do I do believe that this only good for the current setup which is a private local server within my computer. If I want expand my servers I definitely need to up my scripts and set up routine cron jobs to monitor my server activity especially for things like web server traffic and for if I want to expand my server and services for linux administration. Alternatively I could try to use cron or scripts to utilize my firewalls on both of my servers to detect malicious activty

<https://www.getastra.com/blog/security-audit/web-server-security/>

<https://www.cisco.com/site/us/en/learn/topics/security/what-is-user-security.html>

<https://ubuntu.com/server/docs/security-firewall>

**Sources**

<https://askubuntu.com/questions/61826/how-do-i-check-the-sha1-hash-of-a-file>

<https://superuser.com/questions/98825/how-to-check-if-a-directory-exists-in-linux-command-line>

<https://serverfault.com/questions/430895/finding-latest-successful-logins-and-failed-attempts-to-a-centos-server>