AUV

Bandit – Aiklavyaveer

Level 0:

- used man ssh command and read that ssh username@host -p port and then entering password when prompted allows to login

Level 0-1:

- read ls, cd, cat, file, du, find command descriptions
- used ls to see all files and cat to see content in readme file

Level 1-2:

- learnt that for filenames beginning with a -, we can give the whole directory ./-file to make it work like normal or else it thinks that the - is a part of some command

Level 2-3:

- filename contained spaces, so entered the name within single quotes

Level 3-4:

- ls to see contents, cd into inhere directory
- used ls -a to see even hidden files, as ls did not show hidden files
- found hidden file, used cat to display contents

Level 4-5:

- used file ./* command to check all file types
- all files were of type data except one which was of type ASCII text that contained the password.

Level 5-6:

- used find ./ -type f -size 1033c to find all files inside the current directory that are of type file and match a size of 1033 bytes (c=bytes)

Level 6-7:

- used find / -type f -user bandit7 -group bandit6 -size 33c where the / in find / directs it to search from the root or search the entire system

Level 7-8:

- learnt about the grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd and also awk commands
- grep uses regex to filter for matching patterns from a file
- used grep 'millionth' data.txt to match the word millionth and print the line it is contained in
- (in this case) can also add | awk '{print $2}' to take the output from grep command as input into awk and print the 2nd column in the line containing millionth, ie only print the password
    - awk '/millionth/ {print $2}' data.txt also works

Level 8-9:

- used sort data.txt | uniq -c to sort and remove duplicates, then find the number of occurrences of each (now unique) line
- or can use sort data.txt | uniq -u that only shows the unique lines
- using sort first is necessary because uniq only compares adjacent lines for uniqueness

Level 9-10:

- used strings data.txt to extract all human readable strings then combined | grep '=' to filter for those containing '='