**Linux for hackers 7 – Managing user environment variables**

1)Shell variables 2) environment variables

Environment variables are system wide variables built into your system and interface that control the way your system looks, acts and feels to the user and they are inherited by any child shells or processes.

Shell variables are typically listed in lowercase & are only valid in the shell they are set in.

\*\* variables are simply strings in key:value pairs. It looks like KEY=value

KEY=value1:value2

**$env**

\*\*environment variables are uppercase

$set | more

$set | grep HISTSIZE

**Changing variable values for a session:** $HISTSIZE = 0

HISTSIZE(stores last 1000 commands) is the history to store past commands, this will not leave any evidence on the target system after running commands. **$HISTSIZE = 0**

**Making variable value changes permanent: $echo HISTSIZE > ~/valueofhistsize.txt**

By running the above command we are taking the backup of the file

If you want to take the backup which makes more sense we can try something like this

**$echo HISTSIZE > ~/11072022histsize.txt**

Let’s say you had made some changes to the file or made a change to the variable, we can make the change permanent by entering **export** and then the name of the variable we had changed.

**$export HISTSIZE**

Now out HISTSIZE is 0 and if we want to reset it to 1000 we can run the below command.

**$HISTSIZE=1000**

**$export HSITSIZE**

**Changing your shell prompt:** shell prompt is a environment variable that gives information such as user you are operating and the directory in which you’re currently working, default shell prompt in kali takes the following format. ***username@hostname:current\_directory***

Ex: for root user it will be **root@kali:current\_dir**

Using **PS1** variable we can change the prompt. The **PS1** variable has as set of placeholders for information you want to display in the prompt, including the following:

\u The name of current user

\h The hostname

\w The base name of current working directory

This is useful while running multiple shells/accounts, who you are and where you are & the current system is

**$PS1=”world’s best hacker: #”**

**World’s best hacker: #**

**$export PS1 –** will make changes to other sessions as well.

If you want your terminal to look like windows command prompt use \w

**$export PS1=’C:\w’**

**C:/tmp**

**\*\*Changing your path:** PATH variable which controls where on your system your shell will look for commands you enter such as cd, ls, cat, …

Most commands located in sbin or bin subdirectory like **/usr/local/sbin or /usr/local/bin**

If bash doesn’t find the command in one of these directories in your PATH variable, it will give error **“command not found”** even if it is there in your directory but not in your PATH variable.

To see which directories are stored in PATH variable by using $**echo $PATH**

**\*\*Adding to the PATH variable :** To add newhackingtool to your PATH variable enter the following:

**$PATH = $PATH:/root/newhackingtool**

**$echo $PATH**

/usr/local/sbin:usr/local/bin:/usr/sbin:/sbin/bin:/root/newhackingtool

(to check if the newhackingtool is added to the PATH)

* IMP: DO NOT add more directories to your path variable as the system have to search through all the directories in the PATH to find commands which could slow down your system and hacking.

\*\*How **NOT** to add to PATH variable

$PATH=/root/newhackingtool

kali >echo $PATH

/root/newhackingtool

\*\*Creating a user defined variable : custom variable

**$MYNEWVARIABLE = “hacking is the most valuable skill in the 21st century”**

This assigns a string to the variable **MYNEWVARIABLE**

**$echo $MYNEWVARIABLE**

**\***Just like system environment variables, user defined variables must be exported to persist to new sessions. To delete this new variable or any variable use **‘unset’** command. Make sure you are deleting a system variable, though because your system will operate much different afterwards

$unset $MYNEWVARIABLE

**Exercise**

1. View all of your environment variables with the more command.

2. Use the echo command to view the HOSTNAME variable.

3. Find a method to change the slash (/) to a backslash (\) in the faux Microsoft cmd PS1example

4. Create a variable named MYNEWVARIABLE and put your name in it.

5. Use echo to view the contents of MYNEWVARIABLE.

6. Export MYNEWVARIABLE so that it’s available in all environments.

7. Use the echo command to view the contents of the PATH variable.

8. Add your home directory to the PATH variable so that any binaries in your homedirectory can be used in any directory.

9. Change your PS1 variable to “World’s Greatest Hacker:”.