OverTheWire: Bandit – All Levels

Process	How (commands/notes/actions taken)	Outcome (what happened as a result)
Install SSH	My linux had no ssh pre-installed so I followed: https://linuxconfig.org/how-to-install-ssh-secure-shell-service-on-kali-linux and installed ssh `sudo apt install ssh` `sudo systemctl enable ssh`	ssh installed
Level 0 → Level 1	ssh <username>@<remote> -p<portnumber> `ssh bandit0@bandit.labs.overthewire.org -p 2220` `ls -all -s` `cat readme` **Flag obtained: NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL**	Connected to the bandit server as user bandit0 and password bandit0 ls showed what files are in the current working directory. Viewed contents of file 'readme' and obtained password.
Level 1 → Level 2	Exit from Bandit0: `exit` `ssh bandit1@bandit.labs.overthewire.org -p 2220` `ls` `cat ./-` **Flag obtained: rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi**	Dashed Filename: have to specify the full location of the file from the current working directory. Added ./ in front of filename.
Level 2 → Level 3	Exit from Bandit1: `exit` `ssh bandit2@bandit.labs.overthewire.org -p 2220`	Spaces in Filename: · Type command, space, first word of filename and tab OR

	ls **Flag obtained: aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG**	· Type command, space, filename but before every space, insert "\"
Level 3 → Level 4	Exit from Bandit2: exit ssh bandit3@bandit.labs.overthewire.org -p 2220 ls ls -a cd inhere ls -a cat .hidden **Flag obtained: 2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe**	Hidden Files: Include -a after command to include hidden files.
Level 4 → Level 5	Exit from Bandit3: exit ssh bandit4@bandit.labs.overthewire.org -p 2220 ls cd inhere ls cat ./-file00 (gave gibberish) cat ./-file01 (gave gibberish) cat ./-file02 (gave gibberish)	2 methods: • Open 1 by 1 • Use file ./*

	cat ./-file03 (gave gibberish) cat ./-file04 (gave gibberish) cat ./-file05 (gave gibberish) cat ./-file06 (gave gibberish) cat ./-file07 (gave readable text: password) --- file ./* (gives the type of data inside the file, look for ASCII) **Flag obtained: lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR**	
Level 5 → Level 6	Exit from Bandit4: exit ssh bandit5@bandit.labs.overthewire.org -p 2220 ls cd inhere ls find -size 1033c ls -l ./maybehere07/.file2 cat ./maybehere07/.file2 ---	There is only 1 file that is 1033 bytes in size. Verify the file with ls -l: -rw-r----- 1 root bandit5 1033 Apr 23 18:04 ./maybehere07/.file2 The file is indeed non executable OR Specify all criterias in the find command. (c stands for bytes)

	find ./* -size 1033c -type f ! -executable cat ./maybehere07/.file2 **Flag obtained: P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU**	
Level 6 → Level 7	Exit from Bandit5: exit ssh bandit6@bandit.labs.overthewire.org -p 2220 ls find / -type f -user bandit7 -group bandit6 -size 33c (too many Permission Denied errors) find / -type f -user bandit7 -group bandit6 -size 33c 2>/dev/null (excluded all errors) cat /var/lib/dpkg/info/bandit7.password **Flag Obtained: z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S**	Error exclusion learnt from: https://www.cyberciti.biz/faq/bash-find-exclude-all-permission-denied-messages/
Level 7 → Level 8	Exit from Bandit6: exit ssh bandit7@bandit.labs.overthewire.org -p 2220 ls grep millionth data.txt **Flag Obtained: TESKZC0XvTetK0S9xNwm25STk5iWrBvP**	Grep <word> <file> Gives the entire line with the specified word.
Level 8 → Level 9	Exit from Bandit7: exit	One limitation of the uniq command is that it will only identify duplicates that are adjacent, or next to each other, in the file.

	ssh bandit8@bandit.labs.overthewire.org -p 2220 ls sort data.txt | uniq -u **Flag Obtained: EN632PlfYiZbn3PhVK3XOGSlNInNE00t**	Thus, we will need to sort the file first before using uniq.
Level 9 → Level 10	Prev flag: EN632PlfYiZbn3PhVK3XOGSlNInNE00t ssh bandit9@bandit.labs.overthewire.org -p 2220 ls grep == data.txt grep -a == data.txt **Flag Obtained: G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s**	Doing grep without any flags gave: grep: data.txt: binary file matches. Adding -a or –text flag processes a binary file as if it were text because the txt file contains random binary bytes Highlighted 4 lines with multiple consecutive '=' Picked the correct looking one.
Level 10 → Level 11	Exit from Bandit9: exit ssh bandit10@bandit.labs.overthewire.org -p 2220 ls base64 --decode data.txt **Flag Obtained: 6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM**	Reference: https://www.serverlab.ca/tutorials/linux/administration-linux/how-to-base64-encode-and-decode-from-command-line/
Level 11 → Level 12	Exit from Bandit10: exit	

	`ssh bandit11@bandit.labs.overthewire.org -p 2220` `ls` `cat data.txt | tr 'a-zA-Z' 'n-za-mN-ZA-M'` **Flag Obtained: JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv**	ROT13 (rotate by 13 spaces) A → N, a→n. Encryption and decryption goes the same way.
Level 12 → Level 13	Exit from Bandit11: `exit` `ssh bandit12@bandit.labs.overthewire.org -p 2220` `ls` `mkdir /tmp/Agnes` `cd /tmp/Agnes` `cp ~/data.txt ./` `mv data.txt dataHex.txt` Decompression 1: `xxd -r dataHex.txt dataComp.txt` `cat dataComp.txt` Rename dataComp to gzip file `mv dataComp.txt dataComp.gz` `gzip -d dataComp.gz` `cat dataComp` (still unreadable)	Create a directory where I have permission to create files revert the hexdump using xxd and get the actual data which is not readable To figure out what decompression we need to use, look at the first bytes in the hexdump to find the file signature. (1fb8) https://en.wikipedia.org/wiki/List_of_file_signatures shows that 1f8b is a gzip compressed file.

	Decompression 2: `xxd dataComp` `mv dataComp dataComp.bz2` `bzip2 -d dataComp.bz2` `cat dataComp (still unreadable)` Decompression 3: `xxd dataComp` `mv dataComp dataComp.gz` `gzip -d dataComp.gz` `cat dataComp (still unreadable)` Decompression 4: `xxd dataComp (too many lines)` `xxd dataComp | head` `mv dataComp dataComp.tar` `tar -xf dataComp.tar` `ls` `xxd data5.bin | head`	File is still compressed. Repeat process. 425a 68: Compressed file using Bzip2 algorithm version 2 (bz2) File is still compressed. Repeat process. File is archived this time. | head gives me only the first 10 lines in the file which is easier to see since I only need to know the name of the archive file. use tar to extract the file.

	`tar -xf data5.bin` `ls` Decompression 5: `xxd data6.bin` `bzip2 -d data6.bin` `ls` Decompression 6: `xxd data6.bin.out | head` `tar -xf data6.bin.out` `ls` Decompression 7: `xxd data8.bin | head` `mv data8.bin data8.gz` `gzip -d data8.gz` `ls` `cat data8`	Another archive file data6.bin Another bzip2 file Another archive file data8.bin Another gzip file

	Flag Obtained: wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw	Finally uncompressed the file
Level 13 → Level 14	Exit from Bandit12: exit ssh bandit13@bandit.labs.overthewire.org -p 2220 ls cat sshkey.private ssh -i sshkey.private bandit14@bandit.labs.overthewire.org -p 2220 cat /etc/bandit_pass/bandit14 **Flag Obtained: fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq**	Ssh into bandit14 using the ssh key given in bandit13 (@localhost doesn't work says wrong port used)
Level 14 → Level 15	man nc nc localhost 30000 Enter previous flag. **Flag Obtained: jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt**	Netcat or NC is a utility tool that uses TCP and UDP connections to read and write in a network. Use nc to connect to localhost port 3000 and write the password
Level 15 → Level 16	Exit from Bandit14: exit ssh bandit15@bandit.labs.overthewire.org -p 2220 openssl s_client -connect localhost:30001 Enter previous flag **Flag Obtained: JQttfApK4SeyHwDlI9SXGR50qclOAil1**	openssl s_client is the implementation of a simple client that connects to a server using SSL/TLS Note: adding -ign_eof does not help with anything it seems 😟

Level 16 → Level 17	Exit from Bandit15: `exit` `ssh bandit16@bandit.labs.overthewire.org -p 2220` `nmap -p31000-32000 localhost` `nmap -sV -p31000-32000 localhost` `nmap -sV -T4 -p31000-32000 localhost` PORT STATE SERVICE VERSION 31046/tcp open echo 31518/tcp open ssl/echo 31691/tcp open echo 31790/tcp open ssl/unknown 31960/tcp open echo `openssl s_client -connect localhost:31790` `mkdir /tmp/level16to17` `cd /tmp/level16to17` `cat >> sshkey.txt` Paste the ssh key `chmod 400 sshkey.txt` `ls -l` `ssh -i sshkey.private bandit17@bandit.labs.overthewire.org -p 2220` **Flag Obtained: NIL**	Want to know service as well so I added the -sV flag Added -T4 to speed up the scan. (https://nmap.org/book/performance-timing-templates.html) port 31518 runs only the echo service so we connect to port 31790. create directory and save key in a file. Change permissions to only owner can read only (https://www.pluralsight.com/blog/it-ops/linux-file-permissions) ssh key is given: -----BEGIN RSA PRIVATE KEY----- MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ Ja6Lzb558YW3FZl87ORiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzL0VUYbW JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl J9nOM8OJ0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC

		YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asvlpmS8A vLY9r60wYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama +TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT 8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDi Ttiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu L8ktHMPvodBwNsSBULpG0QKBgBAplTfC1HOnWiMGOU3KPwYWt0O6CdTkmJOmL8Ni blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM 77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b dxviW8+TFVEBl1O4f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9GOtt9JPsX8MBTakzh3 vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY= -----END RSA PRIVATE KEY-----
Level 17 → Level 18	ls man diff diff passwords.new passwords.old **Flag Obtained: hga5tuuCLF6fFzUpnagiMN8ssu9LFrdg**	Diff compares file line by line and by default tells us which lines in one file has to be changed to make the two files identical (aka different lines)
Level 18 → Level 19	Exit from Bandit17: exit ssh bandit18@bandit.labs.overthewire.org -p 2220 ssh -t bandit18@bandit.labs.overthewire.org -p 2220 bash –norc ls cat readme **Flag Obtained: awhqfNnAbc1naukrpqDYcF95h7HoMTrC**	Got logged out with message: “Byebye !” Add bash -norc to bypass .bashrc since that file gets executed right after logging in which is causing the logging out. (Adding /bin/sh at the end works as well.) Check that I have sucessfully logged in.
Level 19 → Level 20	Exit from Bandit18: exit ssh bandit19@bandit.labs.overthewire.org -p 2220	Setuid is a Linux file permission setting that allows a user to execute the file or program with the permission of the owner of that file.

	ls /etc/bandit_pass ls -la -rwsr-x—— 1 bandit20 bandit19 14876 Apr 23 18:04 bandit20-do ./bandit20-do id (check what are the ids of users) ./bandit20-do id (euid is bandit20 means assigned id for me is bandit20) ./bandit20-do cat /etc/bandit_pass/bandit20 **Flag obtained: VxCazJaVykI6W36BkBU0mJTCM8rR95XT**	See which file is the goal List all files and the permissions in current directory. User is bandit20, group is bandit19. This file allows user to run commands as user bandit20 "I am bandit20" if I execute ./bandit20-do Access the file as user bandit20
Level 20 → Level 21	Exit from Bandit19: exit ssh bandit20@bandit.labs.overthewire.org -p 2220 ls echo 'VxCazJaVykI6W36BkBU0mJTCM8rR95XT' | nc -l -p 1234 & (receive the password inputted through echo) ./suconnect 1234 **Flag obtained: NvEJF7oVjkddltPSrdKEFOllh9V1IBcq**	I need to send over the previous flag to a listening port I created Use nc to create a connection in server mode to port 1234 (-p) which listens (-l) for inbound connections. & is used to keep the command running while executing other commands in the same terminal.
Level 21 → Level 22	Exit from Bandit20: exit	

	ssh bandit21@bandit.labs.overthewire.org -p 2220 cd /etc/cron.d/ ls cat cronjob_bandit22 `@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null` `* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null` cat /usr/bin/cronjob_bandit22.sh `#!/bin/bash` `chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv` `cat /etc/bandit_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv` cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv **Flag obtained: WdDozAdTM2z9DiFEQ2mGlwngMfj4EZff**	cronjob runs the /usr/bin/cronjob_bandit22.sh file as bandit22 user read the bash file bash file creates a file in tmp folder and gives read access to everyone (chmod 644) bash file then copies contents in /etc/bandit_pass/bandit22 into the created file in tmp. Read the file in tmp
Level 22 → Level 23	Exit from Bandit21: exit ssh bandit22@bandit.labs.overthewire.org -p 2220 cd /etc/cron.d/ ls cat cronjob_bandit23 cat /usr/bin/cronjob_bandit23.sh	First few steps similar to previous level This bash file set '**myname'** variable as **bandit23** (since this bash script will be run as bandit23, whoami gives

	`` #!/bin/bash myname=$(whoami) mytarget=$(echo I am user $myname \| md5sum \| cut -d ' ' -f 1) echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget" cat /etc/bandit_pass/$myname > /tmp/$mytarget ``` `echo I am user bandit23 | md5sum | cut -d ' ' -f 1` `cat /tmp/8ca319486bfbbc3663ea0fbe81326349` **Flag obtained: QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G**	bandit23). It also sets '**mytarget'** variable as **8ca319486bfbbc3663ea0fbe81326349** (check by running the echo command and replace $myname with bandit23) It then copies what is inside /etc/bandit_pass/bandit22 to tmp/ 8ca319486bfbbc3663ea0fbe81326349
Level 23 → Level 24	Exit from Bandit22: `exit` `ssh bandit23@bandit.labs.overthewire.org -p 2220` `ls /etc/cron.d/` `cat /etc/cron.d/cronjob_bandit24` `cat /usr/bin/cronjob_bandit24.sh`	Bash file is executed as user bandit24 The script executes and deletes all files in the folder '/var/spool/bandit24'. (because they are created by bandit23) **myname**: bandit24 For loop is used to iterate through all the files in the directory (**.*** : All files in current working directory)

```
#!/bin/bash

myname=$(whoami)

cd /var/spool/$myname/foo || exit 1
echo "Executing and deleting all scripts in /var/spool/$myname/foo:"
for i in * .*;
do
    if [ "$i" != "." -a "$i" != ".." ];
    then
        echo "Handling $i"
        owner="$(stat --format "%U" ./$i)"
        if [ "${owner}" = "bandit23" ]; then
            timeout -s 9 60 ./$i
        fi
        rm -rf ./$i
    fi
done
```

mkdir /tmp/Agnes

cd /tmp/Agnes

nano getPass.sh

#!/bin/bash cat /etc/bandit_pass/bandit24 > /tmp/Agnes/flag

cat getPass.sh

touch flag

chmod 777 -R /tmp/Agnes

The first if statement makes sure the directories '.' – bandit24 and '..' -spool are ignored.
owner: information about the owner of the file
If owner is bandit23, current selected file is executed and sent the KILL signal after waiting for 60 seconds and then deleted (-s flag specifies the signal to send to, "9" is an alias for KILL command)

Need to write a script that will be executed from the bandit24 folder (get the password and save in a location that I can access)

Create a folder for my bash file.

Populate my bash file with code to copy contents from the deleted folder to the directory I just created.

Check if code is there.

Create a 'flag' file to put in the flag later.
(touch creates an empty file for use later while cat creates a file for you to immediately populate)

Gives rwx permission to all users for all files in that folder so that bandit24 can execute my script.sh and write into the flag filee

Copy the new bash script into the bandit24 folder for cron job to execute it.

After while, flag file will be populated with the password due to the execution of my script.sh

	cp getPass.sh /var/spool/bandit24/foo cat flag **Flag obtained: VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar** *almost died*	
Level 24 → Level 25	Exit from Bandit23: exit ssh bandit24@bandit.labs.overthewire.org -p 2220 nc localhost 30002 entered: VAfGXJ1PBSsPSnvsjI8p759leLZ9Ggar 0000 mkdir /tmp/Agnes (file exists form previous level) cd /tmp/Agnes nano script25.sh #!/bin/bash >> prevFlag = 'VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar' for first in {0..9}	I am the pincode checker for user bandit25. Please enter the password for user bandit24 and the secret pincode on a single line, separated by a space. Wrong! Please enter the correct current password. Try again. Seems like I need to create a bash script to help me do this for all 10000 combinations of the pincode. *I want to die*

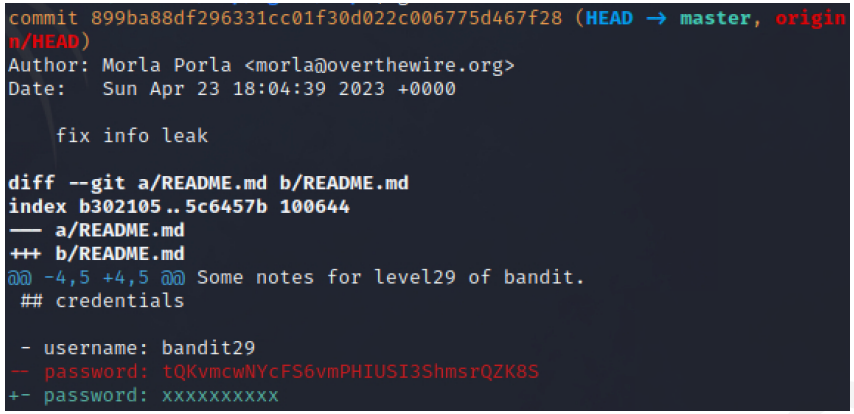
	do for second in {0..9} do for third in {0..9} do for fourth in {0..9} do pin = "$first" + "$second" + "$third" + "$fourth" submit = prevFlag + " " + pin done done done done for i in {0000..9999} → do this instead **Flag obtained:**	
Level 24 → Level 25	Exit from Bandit23: exit	"I am the pincode checker for user bandit25. Please enter the password for user bandit24 and the secret pincode on

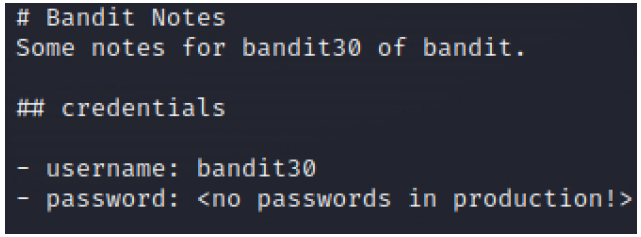
	ssh bandit24@bandit.labs.overthewire.org -p 2220 nc localhost 30002 entered: VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar 0000 mkdir /tmp/Agnes (file exists form previous level) cd /tmp/Agnes nano script25.sh `for i in {0000..9999}` `do` `  echo "VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar $i"` `done | nc localhost 30002` ./script25.sh chmod +x script25.sh ./script25.sh for i in {0000..9999}; do echo "VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar $i"; done | nc localhost 30002 **Flag obtained: p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d**	a single line, separated by a space." Wrong! Please enter the correct current password. Try again. Seems like I need to create a bash script to help me do this for all 10000 combinations of the pin code. My bash script code. Permission denied, need to grant execute permission. Grant permission to execute. Execute bash script (the code gets stuck after a while dk why) Did this instead and worked (but it's the same code leh 😳)
Level 25 → Level 26	Exit from Bandit24: exit ssh bandit25@bandit.labs.overthewire.org -p 2220	There is a ssh key I can use to log into bandit26.

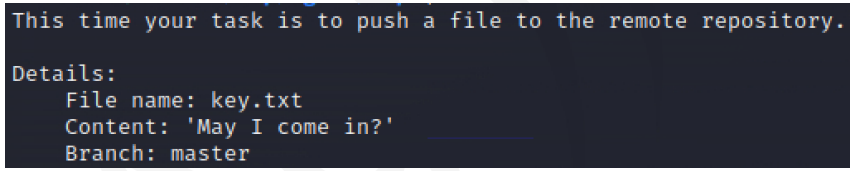
	ls ssh bandit26@bandit.labs.overthewire.org -i bandit26.sshkey -p 2220 cat /etc/passwd (too many lines) grep "bandit26" /etc/passwd bandit26:x:11026:11026:bandit level 26:/home/bandit26:/usr/bin/showtext cat /usr/bin/showtext #!/bin/sh export TERM=linux exec more ~/text.txt exit 0 In vim mode, :set shell=/bin/bash :shell s whoami cat /etc/bandit_pass/bandit26 **Flag Obtained: c7GvcKlw9mC7aUQaPx7nwFstuAIBw1o1**	Seems like I am immediately logged out after logging in. I need to find out where is the default shell for bandit26 (https://www.baeldung.com/linux/find-current-shell) The default shell is /usr/bin/showtext The more command is used to display the contents of a file one screenful at a time. In this case, the text shown is very short and more finishes executing and the next line exit 0 executes which logs me out of bandit26. Solution here is to set the terminal window size small so that more does not execute finish instantly. With that, before more reaches 100%, I press 'V' to enter vim mode. Using vim, I can set the shell to the normal /bin/bash and launch the bash shell (https://superuser.com/questions/287994/how-to-specify-shell-for-vim) Successfully inside bandit26 Get the flag for bandit26

Level 26 → Level 27	Continue from bandit26 (already logged in) `ls` `ls -l` `./bandit27-do cat /etc/bandit_pass/bandit27` **Flag Obtained: YnQpBuifNMas1hcUFk70ZmqkhUU2EuaS**	Binary file bandit27-do is owned by bandit27 and its SETUID bit is set. Meaning if we execute this file together with other commands, we are running those commands as bandit27.
Level 27 → Level 28	Exit from Bandit26: `exit` `ssh bandit27@bandit.labs.overthewire.org -p 2220` `cd /tmp/Agnes` `git clone ssh://bandit27-git@localhost/home/bandit27-git/repo` `git clone ssh://bandit27-git@localhost:2220/home/bandit27-git/repo` `ls` `cd repo` `cat README` **Flag Obtained: AVanL161y9rsbcJIsFHuw35rjaOM19nR**	I should be in a directory that I have permission to create and populate files. Permission denied error due to unknown host (dk why, googled the error) Specified the host manually (https://stackoverflow.com/questions/74289381/trying-to-pass-bandit-level-27-on-overthewire-org-git-clone-not-working)
Level 28 → Level 29	Exit from Bandit27: `exit` `ssh bandit28@bandit.labs.overthewire.org -p 2220`	Same steps as previous level

	cd /tmp/Agnes git clone ssh://bandit28-git@localhost:2220/home/bandit28-git/repo cd repo cat README.md `# Bandit Notes` `Some notes for level29 of bandit.` `## credentials` `- username: bandit29` `- password: xxxxxxxxxx` git log `Author: Morla Porla <morla@overthewire.org>` `Date: Sun Apr 23 18:04:39 2023 +0000` `fix info leak` `commit abcff758fa6343a0d002a1c0add1ad8c71b88534` `Author: Morla Porla <morla@overthewire.org>` `Date: Sun Apr 23 18:04:39 2023 +0000` `add missing data` `commit c0a8c3cf093fba65f4ee0e1fe2a530b799508c78` `Author: Ben Dover <noone@overthewire.org>` `Date: Sun Apr 23 18:04:39 2023 +0000` `initial commit of README.md` git show 899ba88df296331cc01f30d022c006775d467f28	Password censored or smth (knew it -_- *sighs*) Past versions of the file may have the uncensored passwords. Look into git history. Check past commits. “fix info leak” might be to censor the password so means previous version of this fix has the viewable password. Want to see the file before that commit was made. (Commit: red to green)

	 **Flag Obtained: tQKvmcwNYcFS6vmPHIUSI3ShmsrQZK8S**	
Level 29 → Level 30	Exit from Bandit28: exit ssh bandit29@bandit.labs.overthewire.org -p 2220 mkdir git29 cd git29/ it clone ssh://bandit29-git@localhost:2220/home/bandit29-git/repo cd repo cat README.md	Same steps as previous level “production” tells me that there might be branches that are involved in version control.

	 git branch -a git checkout remotes/origin/dev ls cat README.md **Flag Obtained: xbhV3HpNGlTIdnjUrdAlPzc2L6y9EOnS**	Show all branches https://www.atlassian.com/git/tutorials/using-branches/git-checkout
Level 30 → Level 31	Exit from Bandit29: exit ssh bandit30@bandit.labs.overthewire.org -p 2220 cd /tmp/Agnes mkdir git30 cd git30 git clone ssh://bandit30-git@localhost:2220/home/bandit30-git/repo cd repo cat README.md git log	Same steps as previous level “just an epmty file... muahaha” (zzz)

	`git branch -a` `git tag` `git show secret` **Flag Obtained: OoffzGDlzhAlerFJ2cAiz1D41JW1Mhmt**	No commits or branches of interest… Tags are ref's that point to specific points in Git history. Tagging is generally used to capture a point in history that is used for a marked version release (https://www.atlassian.com/git/tutorials/inspecting-a-repository/git-tag) There is a secret file.
Level 31 → Level 32	Exit from Bandit30: `exit` `ssh bandit31@bandit.labs.overthewire.org -p 2220` `cd /tmp` `mkdir Agnes` `cd Agnes` `git clone ssh://bandit31-git@localhost:2220/home/bandit31-git/repo` `cd repo` `cat README.md`  `cat >> key.txt` `May I come in?` `ls -l -a`	Same steps as previous level Need to create a file called key.txt with contents: “May I come in?”

	`cat .gitignore` `git add -f key.txt` `git commit` `git push` **Flag Obtained: rmCBvG56y58BXzv98yZGdO7ATVL5dW8y**	.gitignore lists all file type that git will not push to the repository. In this case, .txt files. I need to use -f flag to force add txt files Commit changes
Level 32 → Level 33	Exit from Bandit31: `exit` `ssh bandit32@bandit.labs.overthewire.org -p 2220` `ls` `pwd` `sh: 1: PWD: Permission denied` `$SHELL` `$0` `whoami` `cat /etc/bandit_pass/bandit33` **Flag Obtained: odHo63fHiFqcWWJG9rLiLDtPm45KzUKy**	sh: bourne shell Typing simple commands like ls can't work here. Seems like any commands I type turns into uppercase. Positional arguments are denoted by $0, $1, $2 … (https://www.in-ulm.de/~mascheck/bourne/v7/) $0 is sh "WELCOME TO THE UPPERCASE SHELL". Need to switch to bourne shell aka sh which is at position 0 Shows that I am user bandit33 which means I can cat into the password file.

~ The End ~