**Bandit Solution**

Bandit0

First, I wanted to display the folder’s files. For doing so I entered the command **ls**. There was a file called **“readme”**.

For displaying file context, I entered the command **cat readme**. The password was inside the file: **boJ9jbbUNNfktd78OOpsqOltutMc3MY1**

Bandit1

Like before I entered the command **ls** for displaying folder’s file. There was a file called **“–“.**

For displaying the file context, I entered the command **cat -**. Nothing was displayed, but in the same it didn’t finish reading the file. When I type some letters and press enter, they were printed again.

I also entered the command **file –** for displaying file type, and after I type some letters and pressed enter, I saw that the file type is: **/dev/stdin: ASCII text**. After reading <https://unix.stackexchange.com/questions/16357/usage-of-dash-in-place-of-a-filename> I understand that the cat command treat – as /dev/stdin file, which is the standard input file.

For displaying the file (and not stdin), I entered the command **ls ./-**. The password was inside the file: **CV1DtqXWVFXTvM2F0k09SHz0YwRINYA9**

Bandit2

First, I wanted to display the folder’s files. For doing so I entered the command **ls**. There was a file called **“spaces in this filename”**.

For displaying the file context, I entered the command **ls “spaces in this filename”** (with the quotation marks). The password was inside the file: **UmHadQclWmgdLOKQ3YNgjWxGoRMb5luK**

Bandit3

For displaying folder context, I entered the command **ls**. There was a directory named **“inhere”**.

I entered the directory using the command **cd inhere**.

After that I used the **ls** command again, and I saw nothing. I used the command **ls -la** for displaying all files and sub directories and I saw there was a hidden file named **“.hidden”**.

For displaying the file context I entered the command **cat .hidden** and the password was there: **pIwrPrtPN36QITSp3EQaw936yaFoFgAB**

Bandit4

**ls** command for displaying the folder context. I saw there was a directory named “inhere”.

**cd inhere** for entering the directory.

**ls** again for displaying all directory’s files.

There was 10 files: **-file00, -file01, -file02, ….., -file09**.

In the mission description they said that the password is inside a readable file (meaning inside an ASCII Text file). I entered the command **file ./\*** for displaying all files types. The only readable file was **-file07**.

For displaying the password, I entered the command **cat ./-file07**: **koReBOKuIDDepwhWk7jZC0RTdopnAYKh**

Bandit5

In the description they said that the password’s file:

* Human readable
* 1033 bytes in size
* Not executable

After reading inside the find man page (using the command **man find**) and searching within it (using /<word to search>) I found out that for finding the wanting file I need to enter the following command: **find -readable -size 1033c \! -executable -print** .The output was: **“./maybehere07/.file2”**.

For displaying the password, I entered the command **cat** **./maybehere07/.file2**: **DXjZPULLxYr17uwoI01bNLQbtFemEgo7**

Bandit6

In the description they said that the password’s file:

* Can be anywhere in the server
* Owned by user bandit7
* Owned by group bandit6
* 33 bytes in size

After reading inside the find man page (using the command **man find**) and searching within it (using /<word to search>) I found out that for finding the wanting file I need to enter the following command: **find / -user bandit7 -group bandit6 -size 33c 2>/dev/null**. The output was: **“/var/lib/dpkg/info/bandit7.password”**.

[**2>/dev/null** is for not displaying the find error message like permissions]

For displaying the password, I entered the command **cat /var/lib/dpkg/info/bandit7.password**: **HKBPTKQnIay4Fw76bEy8PVxKEDQRKTzs**

Bandit7

For finding the line that contains the word millionths I put the following command:

**cat data.txt | grep millionths,** and got the following output: **millionth cvX2JJa4CFALtqS87jk27qwqGhBM9plV**

Bandit8

For finding the only line that occurs only once I entered the command:

**cat data.txt | uniq -u**, and got the following output: **UsvVyFSfZZWbi6wgC7dAFyFuR6jQQUhR**

Bandit9

For displaying all the lines that start with several ‘=’ we need to use **grep –regexp=^=\***

For displaying all readable lines we need to use the command **strings data.txt**

To get the password we need to pie these commands: **strings data.txt | grep –regexp=^=\***, and we can see the password:

========== password

========== isa

========== **truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk**

Bandit10

For decoding the file, I entered the command: **base64 -d data.txt** and got the output: **The password is IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR**

Bandit11

For displaying the file context, we need to enter the command: **cat data.txt**.

After reading the link <https://en.wikipedia.org/wiki/ROT13>, I understood that for doing ROT-13 I need to enter the command: **tr ‘N-ZA-Mn-za-m’ ‘A-Za-z’.**

After piping the command, we can get the password: **cat data.txt |** **tr ‘N-ZA-Mn-za-m’ ‘A-Za-z’.**

The command output was: **The password is 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu**

Bandit12

First, I created a folder in /temp and copy the file there:

**mkdir /tmp/samuel199**

**cp data.txt /tmp/samuel199**

**cd /tmp/samuel199**

First, to convert the hex-dump to binary I entered the command: **xxd -r data.txt data.bin**

Then, for decompress it I needed to how it was compressed.

I entered the command: **file data.bin** and got the following output: **”data.bin: gzip compressed data, was "data2.bin", last modified: Tue Oct 16 12:00:23 2018, max compression, from Unix”**

I change the file extension to gzip using the command: **mv data.bin data.gz**, and then I decompress it using the command: **gzip -d data.gz**. Output file name is: **data.**

Same like before, I entered the command: **file data** and got the following output: “**data: bzip2 compressed data, block size = 900k”**

I decompress the file using the command: **bzip2 -d data**, and got output file named: **data.out**

I entered the command: **file data.out** and got the following output: **“data.out: gzip compressed data, was "data4.bin", last modified: Tue Oct 16 12:00:23 2018, max compression, from Unix”**

I change the file extension to gzip using the command: **mv data.out data.gz**, and then I decompress it using the command: **gzip -d data.gz**. Output file name is: **data.**

I entered the command: **file data** and got the following output: **“data: POSIX tar archive (GNU)”**

I decompress the file using the command: **tar -x -f data**, and got output file named: **data5.bin**

I entered the command: **file data5.bin** and got the following output: **“data5.bin: POSIX tar archive (GNU)”**

I decompress the file using the command: **tar -x -f data5.bin**, and got output file named: **data6.bin**

I entered the command: **file data6.bin** and got the following output: **“data6.bin: bzip2 compressed data, block size = 900k”**

I decompress the file using the command: **bzip2 -d data6.bin**, and got output file named: **data6.bin.out**. I changed the name to data by using the command: **mv data6.bin.out data**

I entered the command: **file data** and got the following output: **“data: POSIX tar archive (GNU)”**

I decompress the file using the command: **tar -x -f data**, and got output file named: **data8.bin**

I entered the command: **file data8.bin** and got the following output: **“data8.bin: gzip compressed data, was "data9.bin", last modified: Tue Oct 16 12:00:23 2018, max compression, from Unix”**

I change the file extension to gzip using the command: **mv data8.bin data8.gz**, and then I decompress it using the command: **gzip -d data8.gz**. Output file name is: **data8.**

I entered the command: **file data8** and got the following output: **“data8: ASCII text”**

For displaying the password, I entered the command **cat data8**, and got the output: **“The password is 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL”**

Bandit13

First, for displaying folder’s files, I entered the command: **ls**. I saw there was only one file: **“sshkey.private”.**

I used the command **cat sshkey.private** for displaying the file context, and I got the following output:

**-----BEGIN RSA PRIVATE KEY-----**

**MIIEpAIBAAKCAQEAxkkOE83W2cOT7IWhFc9aPaaQmQDdgzuXCv+ppZHa++buSkN+**

**gg0tcr7Fw8NLGa5+Uzec2rEg0WmeevB13AIoYp0MZyETq46t+jk9puNwZwIt9XgB**

**ZufGtZEwWbFWw/vVLNwOXBe4UWStGRWzgPpEeSv5Tb1VjLZIBdGphTIK22Amz6Zb**

**ThMsiMnyJafEwJ/T8PQO3myS91vUHEuoOMAzoUID4kN0MEZ3+XahyK0HJVq68KsV**

**ObefXG1vvA3GAJ29kxJaqvRfgYnqZryWN7w3CHjNU4c/2Jkp+n8L0SnxaNA+WYA7**

**jiPyTF0is8uzMlYQ4l1Lzh/8/MpvhCQF8r22dwIDAQABAoIBAQC6dWBjhyEOzjeA**

**J3j/RWmap9M5zfJ/wb2bfidNpwbB8rsJ4sZIDZQ7XuIh4LfygoAQSS+bBw3RXvzE**

**pvJt3SmU8hIDuLsCjL1VnBY5pY7Bju8g8aR/3FyjyNAqx/TLfzlLYfOu7i9Jet67**

**xAh0tONG/u8FB5I3LAI2Vp6OviwvdWeC4nOxCthldpuPKNLA8rmMMVRTKQ+7T2VS**

**nXmwYckKUcUgzoVSpiNZaS0zUDypdpy2+tRH3MQa5kqN1YKjvF8RC47woOYCktsD**

**o3FFpGNFec9Taa3Msy+DfQQhHKZFKIL3bJDONtmrVvtYK40/yeU4aZ/HA2DQzwhe**

**ol1AfiEhAoGBAOnVjosBkm7sblK+n4IEwPxs8sOmhPnTDUy5WGrpSCrXOmsVIBUf**

**laL3ZGLx3xCIwtCnEucB9DvN2HZkupc/h6hTKUYLqXuyLD8njTrbRhLgbC9QrKrS**

**M1F2fSTxVqPtZDlDMwjNR04xHA/fKh8bXXyTMqOHNJTHHNhbh3McdURjAoGBANkU**

**1hqfnw7+aXncJ9bjysr1ZWbqOE5Nd8AFgfwaKuGTTVX2NsUQnCMWdOp+wFak40JH**

**PKWkJNdBG+ex0H9JNQsTK3X5PBMAS8AfX0GrKeuwKWA6erytVTqjOfLYcdp5+z9s**

**8DtVCxDuVsM+i4X8UqIGOlvGbtKEVokHPFXP1q/dAoGAcHg5YX7WEehCgCYTzpO+**

**xysX8ScM2qS6xuZ3MqUWAxUWkh7NGZvhe0sGy9iOdANzwKw7mUUFViaCMR/t54W1**

**GC83sOs3D7n5Mj8x3NdO8xFit7dT9a245TvaoYQ7KgmqpSg/ScKCw4c3eiLava+J**

**3btnJeSIU+8ZXq9XjPRpKwUCgYA7z6LiOQKxNeXH3qHXcnHok855maUj5fJNpPbY**

**iDkyZ8ySF8GlcFsky8Yw6fWCqfG3zDrohJ5l9JmEsBh7SadkwsZhvecQcS9t4vby**

**9/8X4jS0P8ibfcKS4nBP+dT81kkkg5Z5MohXBORA7VWx+ACohcDEkprsQ+w32xeD**

**qT1EvQKBgQDKm8ws2ByvSUVs9GjTilCajFqLJ0eVYzRPaY6f++Gv/UVfAPV4c+S0**

**kAWpXbv5tbkkzbS0eaLPTKgLzavXtQoTtKwrjpolHKIHUz6Wu+n4abfAIRFubOdN**

**/+aLoRQ0yBDRbdXMsZN/jvY44eM+xRLdRVyMmdPtP8belRi2E2aEzA==**

**-----END RSA PRIVATE KEY-----**

After I read a little about the ssh command, I understand I can log into bandit14 user by using the command and the private ssh key using the following command:

**ssh -i sshkey.private bandit14@localhost**

I got the following output:

**“Could not create directory '/home/bandit13/.ssh'.**

**The authenticity of host 'localhost (127.0.0.1)' can't be established.**

**ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.**

**Are you sure you want to continue connecting (yes/no)?”**

I entered **yes** and then pressed enter - I logged into bandit14 user!

For displaying the next level password, I entered the command: **cat /etc/bandit\_pass/bandit14**. The output for that command was: **4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e.**

Bandit14

First, I tried log in again using the **ssh** command by it didn’t work.

Then I tried to do the same thing with the **telnet** command.

I entered the command: **telnet localhost 30000**

And got the output:

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

After a try, I understand I need to enter a password (with incorrect password I got the output: **“Wrong! Please enter the correct current password”**).

So, I entered the current level password (**4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e**) and got the next level password: **BfMYroe26WYalil77FoDi9qh59eK5xNr**.

Bandit15

I read a little about the command openssl and s\_client (using the **man** command)..

Then for retrieving the password, I entered the command:

**openssl s\_client -connect localhost:30001**.

And then the current level password (**BfMYroe26WYalil77FoDi9qh59eK5xNr**).

The output was: **cluFn7wTiGryunymYOu4RcffSxQluehd.**

Bandit16

For getting all open ports in the server, I entered the command: **nmap -p31000-32000** **localhost**, and got the following output:

**“Starting Nmap 7.40 ( https://nmap.org ) at 2019-09-29 08:48 CEST**

**Nmap scan report for localhost (127.0.0.1)**

**Host is up (0.00020s latency).**

**Not shown: 999 closed ports**

**PORT STATE SERVICE**

**31518/tcp filtered unknown**

**31790/tcp open unknown”**

Then I check the 2 ports (one after the other) and find out that only the port **31790** is using ssl!

I entered the command **openssl s\_client -connect localhost:31790**, entered the current level password (**cluFn7wTiGryunymYOu4RcffSxQluehd**) and got the output:

**“-----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-----"**

I saved the private key locally, using the commands:

**mkdir /tmp/samuelKeys**

**nano /tmp/samuelKeys/sshkey17.private**

And entered the command: **ssh -I /tmp/samuelKeys/sshkey17.private bandit17@localhost**

I got the output:

**“@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@**

**@ WARNING: UNPROTECTED PRIVATE KEY FILE! @**

**@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@**

**Permissions 0644 for '/tmp/samuelKeys/sshkey17.private' are too open.**

**It is required that your private key files are NOT accessible by others.**

**This private key will be ignored.**

**Load key "/tmp/samuelKeys/sshkey17.private": bad permissions”**

I changed the ssh private key file permissions to 0600, using the command:

**chmod 0600 /tmp/samuelKeys/sshkey17.privaye**, and tried again – I logged in successfully.

For not doing the whole process again, I checked what is the bandit17 level’s password using the command: **cat /etc/bandit\_pass/bandit17**, and got the output: **xLYVMN9WE5zQ5vHacb0sZEVqbrp7nBTn**.

[/etc/bandit\_pass contains the levels passwords]

Bandit17

For getting the lines that are different between the files, I entered the command:

**diff passwords.new passwords.old**, and got the output:

**“42c42**

**< kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd**

**---**

**> hlbSBPAWJmL6WFDb06gpTx1pPButblOA”**

Which means that the next level password is: **kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd**

[Be aware you won’t get in, but instead will get **“Byebye !”** as output!]

Bandit18

From the mission description I understand someone modified the **.bshrc** file, and that’s why I can’t log in to user bandit18!

I entered the command: **ssh bandit18@bandit.labs.overthewire.org -p 2220 "cat readme"**,for logging into bandit18 user and then immediately get the context of readme file.

The command’s output was: **IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x**.

Bandit19

After reading a little about **setuid** command, I entered **ls** command for viewing all folder’s files. I saw there was an executable file called: **“bandit20-do”**.

I execute the file using the command **./bandit20-do**, and got the following output:

**“Run a command as another user.**

**Example: ./bandit20-do id**

**“**

First, I tried to enter the command: **./bandit20-do 20**, but got the output: **“env: ‘20’: No such file or directory”**.

Then I realized the binary file expected to get a command as argument.

I entered the command: **./bandit20-do cat /etc/bandit\_pass/bandit20**, and got the output: **GbKksEFF4yrVs6il55v6gwY5aVje5f0j**.

Bandit20

For this exercise I used **tmux**, using the command: **tmux**.

Then I split the window using **ctrl+b** and then **%**.

In the first window I created a server on port 6969 using the command: **nc -l -p 6969**.

And then entered the previous level’s password (**GbKksEFF4yrVs6il55v6gwY5aVje5f0j**).

In the second window I executed the setuid binary using the command: **./suconnect 6969**.

And received the following output:

**“Read: GbKksEFF4yrVs6il55v6gwY5aVje5f0j**

**Password matches, sending next password”**

In the first window the output was: **gE269g2h3mw3pwgrj0Ha9Uoqen1c9DGr**.

Bandit21

First, I read about **cron** and **crontab** using the command: **man cron** and **man crontab(5)**.

Then, I looked into the configuration file using the command: **cat /etc/cron.d/cronjob\_bandit22** and got the output:

**“@reboot bandit22 /usr/bin/cronjob\_bandit22.sh &> /dev/null**

**\* \* \* \* \* bandit22 /usr/bin/cronjob\_bandit22.sh &> /dev/null”**

It means the commands from /usr/bin/cronjob\_bandit22.sh runs at startup and every minute (as user bandit22). It also redirects the output to /dev/null.

For displaying file context of the file cronjob\_bandit22.sh I used the command:

**cat /usr/bin/cronjob\_bandit22.sh** and got the following output:

**“#!/bin/bash**

**chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv**

**cat /etc/bandit\_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv“**

It redirects the context of file /etc/bandit\_pass/bandit22 (bandit22’s password) to the file /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv!!!!!

For displaying the next level password, I entered the command:

**cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv** and got the following output: **Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI**.

Bandit22

I entered the command **cat /etc/cron.d/cronjob\_bandit23**, and got the following output:

**“@reboot bandit23 /usr/bin/cronjob\_bandit23.sh &> /dev/null**

**\* \* \* \* \* bandit23 /usr/bin/cronjob\_bandit23.sh &> /dev/null”**

Then, I looked into the file cronjob\_bandit23.sh using the command:

**cat /usr/bin/cronjob\_bandit23.sh,** and got the following output:

**#!/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**

\*\* myname = whoami = bandit23

\*\* mytaget = the first word in the output of md5sum(“I am user bandit23”)

For checking the md5sum result I created a file with the string using the commands:

**mkdir /tmp/samuelMd5/**

**echo "I am user bandit23" > /tmp/samuelMd5/bandit23.file**

and then I entered the command: **md5sum /tmp/smuelMd5/bandit23.file**, and got the output: **“8ca319486bfbbc3663ea0fbe81326349 /tmp/samuelMd5/bandit23.file”**

For displaying the next level output, I entered the command:

**cat /tmp/8ca319486bfbbc3663ea0fbe81326349**, and got the output:

**jc1udXuA1tiHqjIsL8yaapX5XIAI6i0n**.

Bandit23

I entered the command **cat /etc/cron.d/cronjob\_bandit24**, and got the following output:

**“@reboot bandit24 /usr/bin/cronjob\_bandit24.sh &> /dev/null**

**\* \* \* \* \* bandit24 /usr/bin/cronjob\_bandit24.sh &> /dev/null”**

Then, I looked into the file cronjob\_bandit23.sh using the command:

**cat /usr/bin/cronjob\_bandit24.sh,** and got the following output:

**#!/bin/bash**

**myname=$(whoami)**

**cd /var/spool/$myname**

**echo "Executing and deleting all scripts in /var/spool/$myname:"**

**for i in \* .\*;**

**do**

**if [ "$i" != "." -a "$i" != ".." ];**

**then**

**echo "Handling $i"**

**timeout -s 9 60 ./$i**

**rm -f ./$i**

**fi**

**done**

This script execute all the files that are not . or .. in the folder /var/spool/bandit24.

I created a new file with the following string “cat /etc/bandit\_pass/bandit24 > /tmp/bandit24Password” (copy bandit24’s password to another file) using the command:

**echo “cat /etc/bandit\_pass/bandit24 > /tmp/24pass” > /var/spool/bandit24/samuel.sh**

After a minute I entered the command **cat /tmp/24pass** and got the following output:

cat: /tmp/24pass: Is a directory

with the command **ls /tmp/24pass** I saw there are 2 files inside the folder:

* Level24 = a file that contains the password to the next level
* Script.sh = a file that contains the script that I wrote

For displaying the next level password, I entered the command: **cat /tmp/24pass/level24** and got the output: **UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ**.

Bandit24

First, I needed to create server all possible inputs.

For that I used the commands:

**mkdir /tmp/24Script**

**cd /tmp/24Script**

**nano serverInputCreator.sh**

then, entered the following script:

**#!/bin/bash**

**for i in {0000..5000}**

**do**

**echo "UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ $i" >> input1**

**done**

**for i in {5000..9999}**

**do**

**echo "UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ $i" >> input2**

**done**

changed the file permission (add execute permission) with the command:

**chmod 777 serverInputCreator.sh**

and execute it: **./serverInputCreator.sh**

now the files **input1** and **input2** contains the server’s input.

I entered the commands:

**nc localhost 30002 < input1 >> output**

**nc localhost 30002 < input2 >> output**.

[I redirect all command output to files named **output**].

After several seconds I could enter the command : **grep -v “Wrong!” output** (for displaying all the lines that don’t contains error messages), and got the following output:

**“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.**

**Timeout. Exiting.**

**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.**

**Correct!**

**The password of user bandit25 is uNG9O58gUE7snukf3bvZ0rxhtnjzSGzG”.**

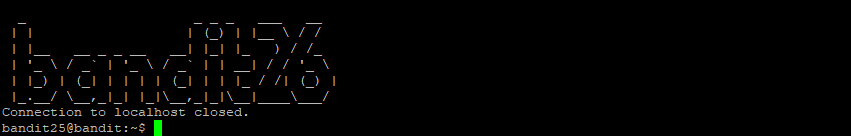
[Be aware: I couldn’t send more than 7000 lines as input to the server, so that’s why I split the input file into 2 input files!]

**[ Notice2: In the internet there is more simple code for solving the level – check it out!]**

Bandit25

First, for displaying the folder’s files I entered the command: **ls**, and got the output: **bandit26.sshkey**.

Then I tried to log into the user with the command: **ssh -i bandit26.sshkey bandit26@localhost** and got the output:

****

Later with little research, I figure out that if I want to know what shell user bandit26 is running I need to enter the command: **grep bandit26 /etc/passwd**, and got the output:

**“bandit26:x:11026:11026:bandit level 26:/home/bandit26:/usr/bin/showtext”**.

I used the command **cat /usr/bin/showtext** to display the file context:

**“#!/bin/sh**

**export TERM=linux**

**more ~/text.txt**

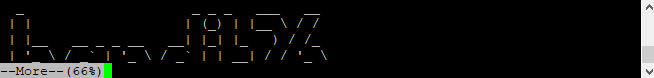
**exit 0”**

It uses **more** command to display text.txt file.

After hours (with little help from google and reading the more manual)

I decided to **resize the terminal screen** (to be smaller) and connect bandit26 with ssh:

**ssh -I bandit26.sshkey bandit26@localhost**. I got the output:

****

{More command filter through text one screenful at a time – so if we decease the size of the screen it will display the file little by little}.

Then **I pressed v** for displaying the file as vim, and the enter **:e /etc/bandit\_pass/bandit26** for editing bandit26’s password file (and of course display it). I got the following output: **5czgV9L3Xx8JPOyRbXh6lQbmIOWvPT6Z**.

We got the password, but if we want to set the shell from showtext to bash, we need to enter (within vim) the command: **:set shell=/bin/bash** and then enter the command **:shell** for running bash shell with user bandit26!

Bandit26

After I got shell, I entered the command **ls** to view the folder’s files.

There was the file **text.txt** and an execute file named **bandit27.do**.

The file bandit27.do used the command **env**, which execute bash files. For getting next level password I created a bash file with the commands:

**mkdir /tmp/samuelScripts**

**cd /tmp/samuelScripts**

**nano bandit26.run**

**chmod 777 bandit26.run**

In the file bandit26.run there was the command: **cat /etc/bandit\_pass/bandit27.**

Then, I entered the command **~/bandit27.do /tmp/samuelScripts/bandit26.run** and got the following output (the password for the next level): **3ba3118a22e93127a4ed485be72ef5ea**.

Bandit27

First before I cloned the repository, I created a directory in tmp:

**mkdir /tmp/samuelGit**

Then I cloned the repository using the command:

**git clone ssh://bandit27-git@localhost/home/bandit27-git/repo /tmp/samuelGit**

I entered the password (3ba3118a22e93127a4ed485be72ef5ea) and got the output:

**“remote: Counting objects: 3, done.**

**remote: Compressing objects: 100% (2/2), done.**

**remote: Total 3 (delta 0), reused 0 (delta 0)**

**Receiving objects: 100% (3/3), done.“**

I viewed the repository files using the command **ls /tmp/samuelGit** and saw there was a single file named **README**. With the command **cat /tmp/samuelGit/README** I got the next level password: **“The password to the next level is:** **0ef186ac70e04ea33b4c1853d2526fa2”**.

Bandit28

Like before:

I created a directory in tmp: **mkdir /temp/samuelGit2**

and cloned the repository using the command:

**git clone ssh://bandit28-git@localhost/home/bandit28-git/repo /tmp/samuelGit2**

I entered the password (0ef186ac70e04ea33b4c1853d2526fa2) and got the output:

**“remote: Counting objects: 9, done.**

**remote: Compressing objects: 100% (6/6), done.**

**remote: Total 9 (delta 2), reused 0 (delta 0)**

**Receiving objects: 100% (9/9), done.**

**Resolving deltas: 100% (2/2), done.”**

I entered the command **cd /tmp/samuelGit3** and viewed the repository files using the command **ls** and saw there was a single file named **README.md**. With the command **cat README.md** I got the output:

**“# Bandit Notes**

**Some notes for level29 of bandit.**

**## credentials**

**- username: bandit29**

**- password: xxxxxxxxxx”**.

After several tries, I decided to check the history of the commits by entering the command:

**git log**, and got the output:

**“commit 073c27c130e6ee407e12faad1dd3848a110c4f95**

**Author: Morla Porla <morla@overthewire.org>**

**Date: Tue Oct 16 14:00:39 2018 +0200**

**fix info leak**

**commit 186a1038cc54d1358d42d468cdc8e3cc28a93fcb**

**Author: Morla Porla <morla@overthewire.org>**

**Date: Tue Oct 16 14:00:39 2018 +0200**

**add missing data**

**commit b67405defc6ef44210c53345fc953e6a21338cc7**

**Author: Ben Dover <noone@overthewire.org>**

**Date: Tue Oct 16 14:00:39 2018 +0200**

**initial commit of README.md“**

Maybe the last commit was to hide the password! To display the last commit’s changes, I entered the command: **git show 073c27c130e6ee407e12faad1dd3848a110c4f95**, and got the following output:

**“commit 073c27c130e6ee407e12faad1dd3848a110c4f95**

**Author: Morla Porla <morla@overthewire.org>**

**Date: Tue Oct 16 14:00:39 2018 +0200**

**fix info leak**

**diff --git a/README.md b/README.md**

**index 3f7cee8..5c6457b 100644**

**--- a/README.md**

**+++ b/README.md**

**@@ -4,5 +4,5 @@ Some notes for level29 of bandit.**

**## credentials**

**- username: bandit29**

**-- password: bbc96594b4e001778eee9975372716b2**

**+- password: xxxxxxxxxx”**

[Which means that the next level’s password is: **bbc96594b4e001778eee9975372716b2**.]

Bandit29

Before I cloned the repository, I created a directory in tmp:

**mkdir /tmp/samuelGit3**

Then I cloned the repository using the command:

**git clone ssh://bandit29-git@localhost/home/bandit29-git/repo /tmp/samuelGit3**

I entered the password (bbc96594b4e001778eee9975372716b2) and got the output:

**“remote: Counting objects: 16, done.**

**remote: Compressing objects: 100% (11/11), done.**

**remote: Total 16 (delta 2), reused 0 (delta 0)**

**Receiving objects: 100% (16/16), done.**

**Resolving deltas: 100% (2/2), done.“**

I entered the command **cd /tmp/samuelGit3** and viewed the repository files using the command **ls** and saw there was a single file named **README.md**. With the command **cat README.md** I got the output:

**“# Bandit Notes**

**Some notes for bandit30 of bandit.**

**## credentials**

**- username: bandit30**

**- password: <no passwords in production!>”**

With git branch –all I saw there are other branches in origin:

**“\* master**

**remotes/origin/HEAD -> origin/master**

**remotes/origin/dev**

**remotes/origin/master**

**remotes/origin/sploits-dev”**

With **git checkout <branchName>** I could switch to other branches…

The origin/HEAD and origin/master was the same like our master branch.

But inside origin/dev the README.md file was different:

**“bandit29@bandit:/tmp/samuelGit3/repo$ cat README.md**

**# Bandit Notes**

**Some notes for bandit30 of bandit.**

**## credentials**

**- username: bandit30**

**- password: 5b90576bedb2cc04c86a9e924ce42faf”**

Bandit30

I created a folder with the command **mkdir /tmp/samuelGit4** and cloned the repository using the command: **git clone ssh://bandit30-git@localhost/home/bandit30-git/repo /tmp/samuelGit4**

I entered the password (5b90576bedb2cc04c86a9e924ce42faf) and got the output:

**“remote: Counting objects: 4, done.**

**remote: Total 4 (delta 0), reused 0 (delta 0)**

**Receiving objects: 100% (4/4), done.”**

I moved to the folder (using the command **cd /tmp/samuelGit4**) and displayed Its content using the command **ls**. There was a single file named **README.md** and within it was the string (using the command **cat README.md**: **“just an epmty file... muahaha”**.

After several checks (with branches and commits) I decided to go to the .git folder (you can see it with the command **ls -la**). I saw that inside the file **packed-refs** (using the command

**cat .git/packed-refs**) there are a ref that was not fetched:

**“# pack-refs with: peeled fully-peeled**

**3aa4c239f729b07deb99a52f125893e162daac9e refs/remotes/origin/master**

**f17132340e8ee6c159e0a4a6bc6f80e1da3b1aea refs/tags/secret”.**

Tags from google: “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 (i.e. v1.0.1). A tag is like a branch that doesn't change. Unlike branches, tags, after being created, have no further history of commits.”

With the command **git show f17132340e8ee6c159e0a4a6bc6f80e1da3b1aea** i got the next level password: “**47e603bb428404d265f59c42920d81e5**”.

[Notice: another option to see all refs and tags by using the command: git show-ref]

Bandit31

I created a folder using the command **mkdir /tmp/samuelGit5** and cloned the repository within it: **git clone ssh://bandit31-git@localhost/home/bandit31-git/repo /tmp/samuelGit5**.

I entered the password (47e603bb428404d265f59c42920d81e5) and got the output:

**“remote: Counting objects: 4, done.**

**remote: Compressing objects: 100% (3/3), done.**

**remote: Total 4 (delta 0), reused 0 (delta 0)**

**Receiving objects: 100% (4/4), done.”**

I entered the command **cd /tmp/samuelGit5** and with the command **ls**, I saw there was a single file named **README.md**. With the command **cat README.md** I saw the context of the file:

**“This time your task is to push a file to the remote repository.**

**Details:**

**File name: key.txt**

**Content: 'May I come in?'**

**Branch: master”**

I checked I was in the master branch (using the command **git branch**) and with the command **echo ‘May I come in?’ > key.txt** I created the file.

Then, I tried to push the file using the command git push, but nothing happened (with the command **git status** I saw there is no change!).

I checked the **.gitignore** file (using the command **cat .gitignore**), and saw it ignores all the files with the extension txt!

I removed the .gitignore file using the command **rm .gitignore**, and commited the changes using the command **git commit -a -m “added key.txt file and removed gitignore file”**, pushed It using the command **git push**, and got the following output:

**“Counting objects: 3, done.**

**Delta compression using up to 4 threads.**

**Compressing objects: 100% (2/2), done.**

**Writing objects: 100% (3/3), 312 bytes | 0 bytes/s, done.**

**Total 3 (delta 0), reused 0 (delta 0)**

**remote: ### Attempting to validate files... ####**

**remote:**

**remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.**

**remote:**

**remote: Well done! Here is the password for the next level:**

**remote: 56a9bf19c63d650ce78e6ec0354ee45e**

**remote:**

**remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.**

**remote:**

**To ssh://localhost/home/bandit31-git/repo**

**! [remote rejected] master -> master (pre-receive hook declined)**

**error: failed to push some refs to 'ssh://bandit31-git@localhost/home/bandit31-git/repo'”**

Bandit32

The first thing that caught my eye was the line: “**WELCOME TO THE UPPERCASE SHELL**” when I logged into the user. Every command I enter; converts into uppercase (for example: ls -> LS).

After I read inside the sh manual (using the command **man sh**), I entered **$0** and gain normal shell again (the uppercase-shell is an interactive shell and with the variable $0 we can gain normal shell!).

I enter the command **cat /etc/bandit\_pass/bandit33** for displaying the next level password:

**“c9c3199ddf4121b10cf581a98d51caee”**.

Bandit33

I entered the command **ls** and saw there was a single file named **README**.

With the command **cat README.txt** I got the following output:

**“Congratulations on solving the last level of this game!**

**At this moment, there are no more levels to play in this game. However, we are constantly working**

**on new levels and will most likely expand this game with more levels soon.**

**Keep an eye out for an announcement on our usual communication channels!**

**In the meantime, you could play some of our other wargames.**

**If you have an idea for an awesome new level, please let us know!”**

***GAME SOLVED!***